



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



Aplicació d'estadístiques i resultats de competicions d'*e-sports*

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) – BarcelonaTech

Autor: Marc Badia Romero

Grau en Enginyeria Informàtica

Especialitat en Enginyeria del Software

Directora: Carme Quer Bosor

Departament d'Enginyeria de Serveis i Sistemes d'Informació

Data de defensa: 28 de Gener de 2020

Resum

Els *e-sports* són un fenomen que cada vegada és més popular. Arreu del món, milions de persones gaudeixen d'aquestes competicions.

Per aquest motiu, en aquest projecte s'aspira a desenvolupar una aplicació per a Android orientada a mostrar diverses estadístiques, resultats i informació de diverses competicions, equips i jugadors d'*e-sports*, i que inclou funcionalitats socials com poder comentar les partides. El sistema, a més, consta d'una base de dades on es recullen i actualitzen les dades pertinents a l'aplicació amb el grau més gran d'automatització possible.

Resumen

Los *e-sports* son un fenómeno que cada vez es más popular. En todo el mundo, millones de personas disfrutan de estas competiciones.

Por este motivo, en este proyecto se aspira a desarrollar una aplicación para Android orientada a mostrar diferentes estadísticas, resultados e información de diferentes competiciones, equipos y jugadores de *e-sports*, y que incluye funcionalidades sociales como poder comentar las partidas. El sistema, además, consta de una base de datos donde se recogen y actualizan los datos pertinentes a la aplicación con el mayor grado de automatización posible.

Abstract

E-sports are a phenomenon that is increasingly popular. Around the world, millions of people enjoy these competitions.

For this reason, this project strives to develop an Android app that shows stats, results and information about different *e-sports* competitions, teams and players, and to include social functionalities such as commenting the games. The system also consists of a database where data relevant to the application is collected and updated at the highest possible degree of automation.

Índex

1. Introducció i contextualització	7
1.1. Termes i conceptes	7
1.2. Problema a resoldre	8
1.3. Stakeholders	9
1.3.1. Autor del projecte	9
1.3.2. Directora del projecte	9
1.3.3. Persones interessades en els e-sports	9
1.3.4. Jugadors professionals i staff tècnic	10
1.3.5. Desenvolupadors d'altres aplicacions similars	10
1.3.6. Empreses dels jocs	10
1.4. Organització de la memòria	10
2. Justificació de l'alternativa de resolució escollida	12
2.1. Pàgines web	12
2.1.1. Lolesports	12
2.1.2. Liquipedia	13
2.1.3. Leaguepedia	13
2.1.4. Reddit	13
2.1.5. Eventvods	14
2.1.6. HLTV	14
2.2. Aplicacions mòbil	15
2.2.1. LoL Friends	15
2.2.2. HLTV.org	15
2.2.3. theScore esports	15
2.3. Conclusions	15
3. Abast del projecte	17
3.1. Objectius	17
3.2. Possibles obstacles i solucions	18
3.2.1. Limitació de temps	18
3.2.2. Limitacions de l'API	18
3.2.3. Disponibilitat API	18
3.2.4. Legalitat de les dades	18
4. Metodologia i rigor	19
4.1. Mètode de treball	19
4.2. Eines de seguiment	20
4.3. Mètodes de validació	21
5. Planificació temporal	22
5.1. Descripció i estimació de les tasques	22
5.1.1. Definició d'objectius i requisits	22

5.1.2. Tria i estudi de l'API	22
5.1.3. Disseny de la BD	23
5.1.4. Integració API a BD	23
5.1.5. Disseny de l'aplicació Android	23
5.1.6. Desenvolupament de la REST API	24
5.1.7. Desenvolupament del front-end de l'aplicació Android	24
5.1.8. Fer les crides a la REST API	24
5.1.9. Testing de l'aplicació	24
5.1.10. Desenvolupament de serveis webs per al back-end	24
5.1.11. Gestió del projecte	25
5.2. Taula resum de les tasques	26
5.3. Diagrama de Gantt	27
5.4. Gestió del risc: Plans alternatius i obstacles	28
5.4.1. Reducció de l'abast	28
5.4.2. Limitacions de l'API	28
5.4.3. Legalitat de les dades	28
5.4.4. Disponibilitat API	28
6. Gestió econòmica	29
6.1. Identificació i estimació dels costos	29
6.1.1. Recursos humans	29
6.1.2. Costos genèrics	31
6.1.3. Imprevistos i pla de contingència	32
6.1.4. Pressupost total	33
6.2. Control de gestió	34
7. Especificació de requisits	35
7.1. Procés d'obtenció de requisits	35
7.2 Context	36
7.3. Requisits funcionals	36
7.3.1. Diagrames de casos d'ús	36
7.3.2. Descripció de casos d'ús	41
7.4. Model conceptual de les dades	50
7.4.1. Restriccions d'integritat	52
7.4.2. Descripció de les classes	52
7.5. Requisits no funcionals	54
8. Arquitectura del sistema	56
8.1. Visió global	56
8.2. Disseny del back-end	58
8.2.1. Normalització del diagrama de classes	58
8.2.2. Disseny lògic de la base de dades	61
8.2.3. Disseny de l'API	72
8.3. Disseny del front-end	79

8.3.1. Procés del disseny de la interfície	79
8.3.2. Disseny de la interfície i de la capa de presentació	79
8.3.3. Mapa navegacional	90
8.3.4. Diagrama de classes del disseny	91
8.4. Patrons utilitzats	98
8.4.1. Patró Model-Vista-Controlador	98
8.4.2. Patró singleton	98
8.5. Exemples de diagrames de registre	99
8.5.1. Sign In i registre	99
8.5.2. Canviar llengua	101
8.5.3. Editar nom d'usuari	102
8.5.4. Carregar properes sèries	103
9. Implementació	104
9.1. Tecnologies, llibreries i llenguatges emprats	104
9.1.1. Python	104
9.1.2. MongoDB	105
9.1.3. Java	105
9.1.4. XML	106
9.1.5. AndroidX	106
9.2. Eines de desenvolupament i altres serveis	106
9.2.1. PyCharm	106
9.2.2. Android Studio	107
9.2.3. MongoDB Atlas	107
9.2.4. Heroku	107
9.3. Aspectes rellevants del codi de la implementació	108
9.3.1. Extracció de dades	108
9.3.2. HTML Scraping	109
9.3.3. API	113
9.3.4. Fragments	114
9.3.5. RecyclerView	115
9.3.6. Crides a l'API	117
9.3.7. Google Sign In	117
10. Proves i validació	119
11. Aspectes legals	124
11.1. Lleis aplicables	124
11.2. Llicències	125
12. Seguiment del projecte	127
12.1. Canvis de planificació i pressupost	127
12.2. Canvis de les històries d'usuari	132
13. Informe de sostenibilitat	134

13.1. Autoavaluació	134
13.2. Dimensió econòmica	135
13.3. Dimensió social	135
13.4. Dimensió ambiental	136
13.5. Matriu de sostenibilitat	136
14. Conclusions	137
14.1. Competències tècniques treballades	137
14.2. Treball futur	139
14.3. Conclusions personals	140
15. Referències	142
Annexe 1. Dissenys inicials en paper	147

1. Introducció i contextualització

Aquest projecte és un Treball Final de Grau (TFG) de la Facultat d'Informàtica de Barcelona (FIB) de la Universitat Politècnica de Catalunya (UPC) - BarcelonaTech. Es tracta d'un projecte de modalitat A de l'especialitat d'Enginyeria del Software.

L'objectiu principal d'aquest projecte és dissenyar i desenvolupar una aplicació Android que faciliti la cerca de dades i estadístiques sobre competicions, equips i jugadors de diferents *e-sports* i disposi de diverses funcionalitats socials, com comentar les partides.

1.1. Termes i conceptes

Per començar, en aquesta secció defineixo conceptes necessaris per a la millor comprensió de la memòria.

- **Agile**
El desenvolupament de software *Agile* en el que els requisits i les solucions canvien durant el temps gràcies als equips i als clients. Incentiva respostes ràpides i flexibles als canvis i problemes que puguin sorgir. [1]
- **Android**
Android és un sistema operatiu desenvolupat per Google dissenyat principalment per dispositius mòbils que disposen d'una pantalla tàctil, com *smartphones* i *tablets*. Està basat en Linux i altres software de codi obert. [2]
- **API**
Una API (Application Programming Interface) és un conjunt de rutines i protocols que donen accés a les funcions d'un determinat software. Més concretament per a aquest projecte, una web API s'utilitza per accedir a cert contingut del proveïdor de la API.
- **Campió**
Champion en anglès, com es coneixen els personatges de League of Legends.
- **CSGO**
Diminutiu de Counter Strike Global Offensive, un dels jocs de l'aplicació
- **E-sport**
Un *e-sport* és una competició d'un videojoc multijugador jugada per jugadors professionals. També es pot denominar com a *esport electrònic*, *e-Sport* o *eSport*.
- **Java**
Java és un llenguatge de programació orientat a objectes dissenyat per Oracle en 1996 amb la intenció de que es pugui executar en qualsevol dispositiu. S'utilitza principalment per a aplicacions client-servidor de web.
- **League of Legends**
Un dels jocs de l'aplicació.
- **Mapa (o partida)**

En el context d'un *e-sport*, un mapa es una partida del videojoc en qüestió. Si la partida es divideix en rondes individuals, tot el conjunt de les rondes segueix sent el mateix mapa. El guanyador d'un mapa és el guanyador de la partida.

En el context d'una partida de CSGO, el mapa és l'entorn virtual o escenari on es juga la partida.

- **Scrum**

Scrum és un framework basat en *Agile* dissenyat per equips petits, en el que es divideix la feina en tasques que es completen en diverses iteracions normalment d'entre dos setmanes a un mes, anomenades *sprints*.

- **Sèrie**

En el context d'un *e-sport*, una sèrie és un conjunt de mapes. Solen seguir el format *Best-of*, és a dir, es juguen com a màxim un cert nombre predefinit de mapes i el guanyador de la sèrie és qui ha guanyat la majoria dels mapes. Per exemple, en una sèrie *Best-of 5*, es jugaran un màxim de 5 mapes, i el guanyador serà qui guanyi 3 mapes primer. En alguns casos, pot ser que segueixi el format *Best-of 2*, en el que només es juguen dues partides i la sèrie pot acabar en empat si cada equip guanya una de les partides.

- **Wiki**

Una *wiki* és una pàgina web en la què tots els usuaris poden contribuir afegint i modificant articles. Solen ser d'un tema específic, excepte *Wikipedia*, què és la wiki més popular arreu del món, amb més de 5,5 milions d'articles en anglès i 1,5 milions en castellà. [3]

1.2. Problema a resoldre

Els *e-sports* (també coneguts com a esports electrònics) són competicions de videojocs que es duen a terme entre jugadors professionals. No hi ha una definició clara de què distingeix que un videojoc pugui o no ser un *e-sport*, però normalment són aquells videojocs multijugador (ja siguin en equip o individuals) per als que s'organitzen competicions online o offline (presencials).

Les competicions online de videojocs existeixen des de la dècada dels 70 [4] però cada vegada són més populars gràcies a l'augment de gent amb una bona connexió a Internet, i que, per tant, poden arribar a competir, i a l'augment de la popularitat dels serveis de streaming de vídeo, que fa que més gent sàpiga de l'existència d'aquestes competicions i les gaudeixi. Un altre motiu de l'important augment en popularitat dels *e-sports* és la fàcil accessibilitat a aquestes competicions, un jugador amb el nivell suficient jugant pel seu compte pot arribar a enfrontar-se amb jugadors que ja són professionals, el que pot arribar a fer que els equips s'interessin en ell.

Unes quantes dades sobre la popularitat d'aquest fenomen serien:

- S'estima que 427 milions de persones veuràn algun *e-sport* en el 2019 [5].
- El màxim d'espectadors concurrents en la final del mundial del 2018 de *League of Legends* va ser de més de 200 milions [6].

- La audiència de la *LEC (League of Legends European Championship)* en el 2019 ha augmentat en un 45,4% respecte a la competició europea de l'any anterior i va tenir un pic d'audiència de més de 850.000 espectadors [7].
- La final de la màxima competició espanyola de *League of Legends* supera els 210.000 espectadors en el 2019 [8].
- En el *The International 2019* de *Dota 2*, es van repartir més de 34 milions de dòlars en premis [9].

És indubtable que els *e-sports* han gaudit d'un creixement exponencial en popularitat en els darrers anys. Malgrat això, hi ha poques alternatives a l'hora de buscar informació i estadístiques sobre aquestes competicions, més encara en el mercat mòbil, fet que dificulta molt poder seguir les diferents competicions i estar al dia dels resultats.

Aquest és el problema que aquest projecte pretén resoldre: desenvolupar una aplicació per a Android en la que l'usuari pugui buscar informació i estadístiques de les competicions, partides, equips i jugadors d'*e-sports* que l'interessin, per tal que els usuaris puguin ser capaços de rebre la informació que els interessi fàcilment. Això potencialment també s'aplica als jugadors professionals i a l'staff tècnic dels equips, que podrien utilitzar l'aplicació per consultar dades sobre els seus rivals o ells mateixos.

A més, gairebé cap alternativa ofereix una plataforma de discussió, i les úniques són en anglès. Per això un altre objectiu d'aquesta aplicació és crear una comunitat en la qual els usuaris puguin veure i comentar les partides.

1.3. Stakeholders

1.3.1. Autor del projecte

L'autor del projecte és l'actor més implicat en el projecte, ja que serà qui s'encarregarà de dissenyar, programar i documentar les diferents parts del projecte. A més, com es tracta d'un TFG, es veurà beneficiat per la seva avaluació. L'autor farà tots els rols que hi ha dins d'un equip de desenvolupament de software: cap de projecte, dissenyador, arquitecte, programador de front-end i de back-end, i tester.

1.3.2. Directora del projecte

La directora del projecte, en aquest cas la professora Carme Quer, s'encarrega de supervisar el correcte desenvolupament del projecte.

1.3.3. Persones interessades en els *e-sports*

Totes les persones interessades en aquestes competicions representen el públic objectiu d'aquesta aplicació i, per tant, estan interessades en què el projecte sigui exitós per a poder seguir les competicions d'una manera més fàcil i còmode.

1.3.4. Jugadors professionals i staff tècnic

Com que aquest projecte es centra també en la part d'estadístiques d'aquestes competicions, els jugadors professionals i l'staff tècnic dels equips es poden veure beneficiats de tenir una manera senzilla d'accedir a informació rellevant sobre els rivals i ells mateixos.

1.3.5. Desenvolupadors d'altres aplicacions similars

Com es veurà més endavant en aquest document, les aplicacions i pàgines web de l'estil d'aquest projecte no acaben de complir amb les expectatives del públic general, que busca alternatives. Aquestes empreses possiblement estan interessades a veure si aquest projecte funciona bé per poder millorar les seves aplicacions i pàgines web, o crear-ne de noves.

1.3.6. Empreses dels jocs

Les empreses que han desenvolupat els jocs estan interessades en aquest projecte perquè, al cap i a la fi, ofereix un servei que es basa en el seu producte, donar dades i informació de partides professionals dels seus jocs.

Si l'aplicació és exitosa, els hi va bé, perquè és publicitat gratuïta i, fins i tot, es poden plantejar de crear una aplicació ells mateixos o col·laborar d'alguna manera amb aquest projecte, per tal de facilitar l'obtenció de dades i millorar el servei.

Si, al contrari, l'aplicació és un fracàs i funciona malament o dona informació errònia, és possible que mostrin la seva insatisfacció amb el producte i, en el pitjor cas, es neguin a deixar utilitzar les seves dades a aquest projecte. Aquest últim punt es té en compte en els riscos del projecte.

1.4. Organització de la memòria

La resta d'aquesta memòria s'organitza en els següents apartats:

- **Justificació de l'alternativa de resolució escollida.** Estudi de l'estat de l'art. Anàlisi i comparació de diverses pàgines web i aplicacions mòbil similars a la d'aquest projecte i conclusions sobre la viabilitat del projecte.
- **Abast del projecte.** Especificació dels objectius i dels possibles obstacles del projecte.
- **Metodologia i rigor.** Definició del mètode de treball i dels procediments que es segueixen per dur a terme el seguiment i la validació del projecte.
- **Planificació temporal.** Descripció i estimació temporal de les tasques del projecte i els possibles riscos.
- **Gestió econòmica.** Descripció i estimació dels diferents costos econòmics que componen el projecte.

- **Especificació dels requisits.** Descripció dels requisits funcionals i no funcionals que s'han de satisfer en el sistema resultant i definició del model conceptual de les dades i les classes del sistema.
- **Arquitectura del sistema.** Descripció amb exemples del funcionament intern tant del back-end com del front-end del sistema i raonaments sobre les decisions preses.
- **Implementació.** Descripció de les diferents tecnologies, llibreries, llenguatges i eines de desenvolupament emprades durant el transcurs del projecte i exemples rellevants del codi de la implementació, tant del back-end com del front-end.
- **Proves i validació.** Descripció detallada dels diferents procediments que s'han dut a terme per a la validació del resultat del projecte, és a dir de l'app i sistema resultats..
- **Aspectes legals.** Especificació sobre les diferents lleis i llicències aplicables al sistema que es desenvolupa i raonament sobre la legalitat del mateix.
- **Seguiment del projecte.** Descripció dels canvis en la planificació temporal, el pressupost i les històries d'usuari que s'han fet durant el projecte, a més de raonaments sobre el motiu d'aquests canvis.
- **Informe de sostenibilitat.** Autoavaluació de les diferents dimensions de la sostenibilitat del projecte, i del sistema resultant
- **Conclusions i treball futur.** Conclusions personals sobre la totalitat del projecte i especificació de diferents treballs i millores que es poden fer en l'aplicació resultats en el futur. .
- **Referències.** Webgrafia i referències bibliogràfiques utilitzades en aquesta memòria.

2. Justificació de l'alternativa de resolució escollida

En aquest capítol veurem pàgines web que tenen un objectiu similar al del sistema que es desenvolupa en el projecte (secció 2.1) i aplicacions amb funcionalitats similars a l'aplicació d'aquest projecte (secció 2.2), i analitzarem els seus punts forts i febles (secció 2.3) per tal de decidir si realment cal fer aquest projecte. Algunes d'aquestes pàgines webs i aplicacions són les que s'han usat amb anterioritat al projecte com a fan d'aquestes competicions mentre que d'altres han estat recomanades per amics i/o trobades cercant per Internet.

2.1. Pàgines web

2.1.1. Lolesports

Lolesports [10] és la pàgina oficial de les competicions de *League of Legends*. Permet veure en directe i en diferit les partides de les competicions oficials de *League of Legends* (és a dir, les competicions organitzades per *Riot Games*). En directe, aquestes partides es poden veure des de diferents serveis de streaming de video com *Twitch* i *Youtube* (què estan integrats dins la mateixa web, tal com podem veure a la Figura 1), i en els diferents idiomes que ofereix la mateixa companyia, però en diferit només es poden veure en anglès i en *Youtube* (també en la mateixa web). També permet veure el calendari i la classificació d'aquestes competicions, i permet desactivar els spoilers de partides anteriors.

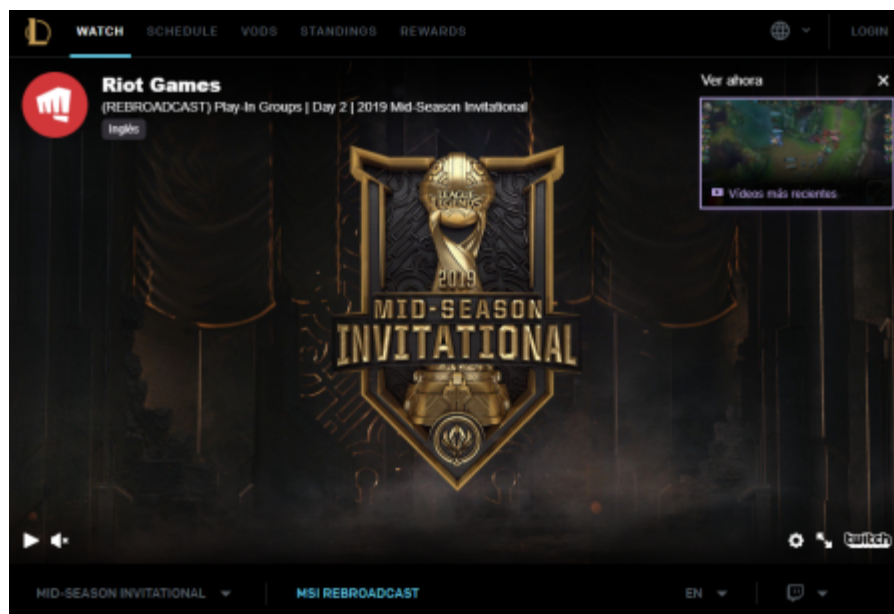


Figura 1. Pantalla principal de lolesports
Font: Captura de pantalla de <https://watch.lolesports.com/>

A més, mostra notícies sobre la teva regió (per exemple, Europa) i s'hi poden comprar entrades per les properes partides, però només proporciona les estadístiques més bàsiques dels equips i jugadors.

Tot i ser la pàgina oficial, fins fa poc no s'utilitzava gaire ja que hi havia altres alternatives millors per mirar les partides, però ha canviat una mica gràcies al fet que donen recompenses dins del joc per utilitzar aquesta pàgina, que en el cas d'aquest projecte, és una fita impossible.

2.1.2. Liquipedia

Liquipedia [11] és un conjunt de *wikis* desenvolupades per *TeamLiquid*, un equip professional d'*e-sports*, què proporcionen informació sobre diferents *e-sports* com *Counter Strike*, *Dota 2*, *Overwatch*, *Hearthstone* i *League of Legends* entre altres.

Al tractar-se d'una wiki tothom pot editar el seu contingut, i dona informació sobre els jugadors, equips, competicions i diferents estadístiques rellevants dels diferents *e-sports*.

Proporciona links a *Twitch* i a *Youtube* per veure les partides i disposa d'un compte enrere per les pròximes partides, però no té cap sistema anti-spoiler.

2.1.3. Leaguepedia

Leaguepedia [12] és una wiki molt similar a *Liquipedia*, però només de *League of Legends*. Disposa pràcticament de les mateixes característiques i funcionalitats que *Liquipedia*, però proporciona moltes més estadístiques tant de les competicions, com dels equips i jugadors. Un altre gran avantatge és que disposa d'informació de pràcticament totes les competicions en la història del *League of Legends*, mentre que *Liquipedia* disposa només de les més recents i importants.

En resum, *Leaguepedia* s'utilitza per a *League of Legends*, però *Liquipedia* s'utilitza per als altres *e-sports*.

2.1.4. Reddit

Reddit [13] és una xarxa social en què es comparteixen i discuteixen posts i imatges. Aquests posts es penegen en diferents *subreddits*. Cada subreddit representa un tema; per exemple, en el subreddit */r/news*, es penegen i es discuteixen notícies, i en el subreddit */r/soccer*, es penegen i es discuteixen posts relacionats amb el futbol.

Tot i que Reddit no és una web explícitament relacionada amb els *e-sports*, en els subreddits relacionats amb els *e-sports*, els usuaris poden discutir les diferents notícies i partides del moment, el que fa que aquests subreddits siguin dignes d'estudiar per poder afrontar aquest projecte.

El punt més fort d'aquestes discussions i el més interessant per a aquest projecte és el sistema de comentaris. Els usuaris poden comentar el post i respondre els comentaris, i altres usuaris els poden votar positivament o negativament, que s'ordenen per puntuació. D'aquesta manera s'evita llegir comentaris poc rellevants a la discussió pertinent o *spam*. Cada subreddit també disposa de moderadors, que creen i fan complir les diferents normes del subreddit (aquestes normes varien entre subreddits). Els usuaris poden ajudar als moderadors reportant posts i comentaris que no compleixen les normes, i els moderadors l'eliminaran si realment no les compleix. En la *Figura 2* es mostra un exemple d'aquest sistema de comentaris.



Figura 2. Sistema de comentaris de Reddit

Font: Captura de pantalla de

https://www.reddit.com/r/leagueoflegends/comments/cvqlda/team_liquid_vs_cloud9_lcs_2019_summer_playoffs/

2.1.5. Eventvods

Originalment aquesta pàgina web va començar com a un subreddit anomenat */r/loleventvods*, on es penjaven links a les partides de les competicions de *League of Legends* intentant evitar els *spoilers* de la millor manera possible. Més endavant, vistes les limitacions d'un subreddit, van crear *Eventvods* [14] amb el mateix propòsit i afegint altres *e-sports* com *Dota 2*, *Counter Strike*, *Overwatch* i *Rocket League*. En la mateixa web es poden veure les partides mitjançant els reproductors integrats de *Twitch* i *Youtube*, similar a *Lol esports*. També ofereixen links a la discussió de la partida en *Reddit* i proporcionen resums d'algunes partides.

2.1.6. HLTV

HLTV [15] és una pàgina web no oficial de *Counter Strike* en què es mostra informació i notícies de les seves competicions, equips i jugadors.

Disposa d'un horari detallat de les properes partides i permet veure estadístiques en directe de les partides que s'estan jugant en el moment. També es poden veure les partides en directe i en diferit en els diferents idiomes disponibles gràcies al reproductor integrat de *Twitch*.

Aquesta pàgina web no disposa d'un sistema anti-*spoilers*, però sí disposa d'un fòrum on els usuaris poden comentar diferents temes.

2.2. Aplicacions mòbil

2.2.1. LoL Friends

LoL Friends és l'aplicació mòbil oficial de *League of Legends*. No hi ha molt a dir d'aquesta aplicació, ja que l'apartat de *e-sports* es limita a poder veure en directe i en diferit les partides de les competicions oficials, com *Lol esports*.

2.2.2. HLTV.org

HLTV.org és l'aplicació mòbil de la pàgina web *HLTV*. Disposa de pràcticament les mateixes funcionalitats que la web original, i és igual que la versió mòbil de la pàgina web.

El seu punt més feble és que és obligatori crear un compte per poder utilitzar l'aplicació.

2.2.3. theScore esports

Aquesta és una aplicació creada per el diari esportiu *theScore*. És clarament l'aplicació mòbil més completa sobre el tema, i proporciona informació i notícies de *League of Legends*, *Dota 2*, *Counter Strike* i *Overwatch*. De cada un d'aquests *e-sports* proporciona estadístiques de les partides, però no dels equips i només les més bàsiques dels jugadors.

Els punts forts són la possibilitat de veure les partides en directe i en diferit mitjançant links a *Youtube*; un sistema anti-*spoiler*, en el que mostra el link a la partida però no el resultat; i el sistema de notifikacions, en el que l'usuari pot rebre notifikacions de les notícies i recordatoris del començament de les partides dels seus equips preferits. Va estar en funcionament fins l'abril del 2019.

2.3. Conclusions

Primer de tot, cal dir que hi ha moltes més pàgines web i aplicacions mòbils que no he inclòs, ja que o bé el que proporcionaven ja ho feia millor una altra web o aplicació que sí que he posat, o bé tenien bones idees però estaven totalment abandonades o desactualitzades. Alguns exemples serien ESL Play, que només proporciona informació sobre competicions organitzades per ESL; o Toornament eSports, que tenia un bon sistema de notifikacions, però està abandonada i serà eliminada en unes setmanes.

Dit això, l'estat actual de pàgines webs relacionades amb els *e-sports* es resumeix en què hi ha bastantes que són molt bones en un cert camp però mediocres en els altres. Per exemple, les wikis són bones per consultar informació i estadístiques, però no permeten comentar les partides i simplement disposen de links a les partides en diferit. Mentrestant, *Reddit* disposa d'un gran sistema de comentaris i *Eventvods* és la millor alternativa per poder veure partides en diferit sense preocupar-se dels *spoilers*.

Com a conclusió, es pot dir que existeixen moltes pàgines bones relacionades amb els *e-sports*, però estan molt especialitzades, i no n'hi ha cap que sigui com a mínim decent en tots els camps que proposa aquest projecte.

Pel que fa a les aplicacions mòbils, l'estat actual és encara pitjor. *LoL Friends* només serveix per veure partides de *League of Legends*. Pel que fa a *theScore e-sports*, gràcies al seu sistema anti-*spoilers* i el sistema de notifikacions, és la millor aplicació mòbil relacionada amb els *e-sports* actualment disponible, però manca en proporcionar estadístiques, una cosa que fa bé *HLTV.org*. El problema d'aquesta última aplicació és que està limitada a *Counter Strike* i, en el fons, és simplement una versió pitjor de la seva pròpia pàgina web.

Com aquest projecte tracta de desenvolupar una aplicació mòbil, ens hem de centrar més a analitzar aquestes. Com s'ha explicat abans, l'única decent es *theScore e-sports*, que ha deixat de funcionar, i així i tot era millorable amb algunes de les idees que proporciona aquest projecte. Com que pertany a una empresa privada, no es pot simplement agafar i millorar amb aquests objectius, pel que la millor solució és dissenyar una de nova.

En resum, opino que hi ha un gran potencial per una aplicació mòbil que pugui ajuntar d'una manera efectiva els diferents camps que fan populars als exemples citats anteriorment i els avantatges que proporcionen els sistemes mòbils.

3. Abast del projecte

En aquest capítol es deixen clars els objectius del projecte (secció 3.1), i els possibles obstacles que podrien donar-se durant el projecte així com les solucions en cas que es produïssin (secció 3.2).

3.1. Objectius

L'objectiu principal d'aquest projecte és desenvolupar una aplicació d'Android en què es mostri informació i estadístiques sobre diferents *e-sports*, a més d'altres funcionalitats com poder comentar les diferents partides. Per a fer-ho, s'utilitzarà Android Studio per a desenvolupar l'aplicació i s'integrarà una API a una base de dades MongoDB per a poder tenir la informació disponible. Per a aquest projecte s'utilitzaran dades de dos e-sports, però es farà de manera que sigui fàcil afegir-ne més en el futur.

Per fer-ho dividim el projecte en tres parts:

- Estudi, tria i integració de l'API. Després d'estudiar les diferents possibilitats de com aconseguir dades sobre els *e-sports*, s'ha arribat a la conclusió que la millor manera és utilitzar una API proporcionada per tercers, ja que no té sentit entrar les dades a mà si ja existeix un sistema de tercers que proporcioni aquestes dades. A més, entrar les dades manualment dificultaria el model de l'aplicació. Llavors, la primera part del projecte serà estudiar les diferents APIs disponibles i triar-ne una. L'elecció es durà a terme llegint les documentacions i fent diverses proves amb les versions gratuïtes. S'escollirà la que més s'apropa als requisits d'aquest projecte, tenint en compte el preu i els requisits legals. Una vegada escollida, s'integrarà en una base de dades pròpia (prèviament dissenyada) per tal de poder afegir informació obtinguda mitjançant altres mètodes.
- Desenvolupament del back-end. Aquesta part del projecte té dos sub-objectius:
 - Completar la informació a la base de dades pròpia. L'API proporcionada per tercers no dona totes les dades necessàries per al projecte. Per aquest motiu, una part important és trobar una manera viable d'obtenir aquestes dades, i aconseguir automatitzar el procés el màxim possible. En cas que sobri temps al finalitzar el projecte, es desenvoluparan uns serveis web per tal de facilitar la gestió d'aquestes dades.
 - Creació d'una REST API. La base de dades pròpia haurà d'estar situada en un servidor. Aquesta base de dades juntament amb l'API que hi donarà accés formaran el back-end de l'aplicació. L'API servirà per comunicar l'aplicació per a mòbils Android amb la base de dades. La millor opció és desenvolupar una REST API. Llavors, en aquesta part, s'estudiaran quines crides són necessàries i s'implementaran.
- Desenvolupament de l'aplicació client d'Android. És la part principal del projecte, en la que es dissenyarà i desenvoluparà una aplicació per a Android on es mostri la

informació recollida, a més d'altres funcionalitats, com integrar una part de xarxa social per tal de poder comentar les partides.

3.2. Possibles obstacles i solucions

3.2.1. Limitació de temps

Aquest projecte és un TFG, pel que hi ha una limitació establerta de temps. Per resoldre aquest problema, es començarà a fer el projecte durant l'estiu i es farà una planificació del temps en la que es tinguin en compte els diferents problemes que puguin sorgir.

3.2.2. Limitacions de l'API

És molt probable que l'API triada no sigui perfecte per al projecte i falti informació important o rellevant. Per a solucionar-ho, s'haurà de buscar diverses maneres de trobar aquesta informació, ja sigui mitjançant tècniques de visió per computador o, en el pitjor cas, afegint-la manualment a la base de dades.

3.2.3. Disponibilitat API

És possible que en algun moment l'API deixi de funcionar durant un temps determinat, probablement menys de 24 hores. En aquest cas no es pot fer molt, ja que durant aquest període de temps no es podran afegir dades noves, però com que la base de dades guarda les dades antigues, es podrà treballar amb aquestes, i l'aplicació seguirà funcionant encara que no tingui les dades més recents.

3.2.4. Legalitat de les dades

Llegint sobre la legalitat de mostrar dades d'equips o jugadors en aplicacions de tercers, s'entén que depèn de la mateixa empresa decidir qui pot utilitzar-les i qui no pot. Per aquest motiu, es contactarà amb l'empresa per demanar permís d'utilitzar les dades. En cas que es denegui el permís, l'aplicació es quedarà com a TFG i no es llançarà al mercat.

4. Metodologia i rigor

En aquest capítol es descriu el mètode de treball usat durant el projecte amb les fases del projecte i que es farà en cada fase (secció 4.1), les eines de seguiment del projecte que s'usaran (secció 4.2) i els mètodes de validació que s'usaran per garantir que el resultat del projecte, és a dir el sistema resultant compleix els objectius i requisits marcats (secció 4.3).

4.1. Mètode de treball

Aquest projecte s'ha dividit en tres fases per tal de poder organitzar la feina d'una manera més efectiva.

En la primera fase s'ha definit el projecte i s'ha marcat la seva conveniència i l'abast, els objectius i la planificació temporal i econòmica. En aquesta fase també es defineixen les diferents funcionalitats de l'aplicació. Aquestes funcionalitats s'estudien per tal de deixar clarament definits els seus requisits i es divideixen en històries d'usuari amb un pes que representa les hores aproximades de feina que comporta la història en qüestió. Aquestes històries a la vegada es divideixen en tasques més petites i concretes. Gràcies a això, es poden dividir les funcionalitats en els diferents sprints de forma més fàcil i equilibrada.

A més, durant aquesta fase s'ha estudiat quins *e-sports* formarien part de l'aplicació, mitjançant enquestes a diferents *fòrums* i xarxes socials i estudiant les aplicacions existents que podien ser rellevants per al projecte. També s'ha estudiat quina API es faria servir per extreure les dades i com s'afegirien les possibles dades que no estiguin disponibles mitjançant l'API.

Al final d'aquesta fase està tot llest per començar a treballar en l'aplicació com a tal.

En la segona fase es tracta de desenvolupar l'aplicació en *Android* i una aplicació servidor per al back-end. S'ha utilitzat una metodologia àgil basada en iteracions similar a *SCRUM* però adaptada per a una sola persona. Les diferents tasques s'han desenvolupat i testejat de forma iterativa en els diferents sprints, de manera que només s'ha fet una tasca a la vegada i no s'ha passat a la següent fins que es comprova que funciona correctament amb els tests adequats.

Pel que fa al back-end, s'han fet i testejat les crides a la base de dades en el moment en què sigui rellevant respecte a la tasca actual. Es fa un *deploy* al servidor cada vegada que una tasca hagi estat completada i testejada.

Per últim, en la tercera fase, es redacta la memòria final del projecte i es prepara la seva defensa. Moltes de les parts d'aquesta memòria es fan en la segona fase i en aquesta s'acaben de redactar i revisar.

4.2. Eines de seguiment

Tot el codi generat s'ha emmagatzemat en un repositori de GitHub per tal d'aconseguir una alta disponibilitat del codi. Aquest repositori té diverses branques per poder provar coses i garantir que la branca màster sempre funciona correctament.

Pel que fa a l'organització de tasques i sprints, es va crear un projecte a l'aplicació Taiga, que permet afegir històries d'usuari i tasques. En aquest projecte s'hi ha afegit la directora del projecte perquè pugui veure el desenvolupament del projecte. Cada història d'usuari té un pes (hores aproximades de feina), una descripció, un conjunt de tasques i, finalment, el temps real de feina que ha comportat.

The screenshot shows a Taiga user story card. At the top, it has the title '#1 Veure properes partides' and the type 'E-SPORTS APP USER STORY'. Below the title, it says 'This User Story belongs to #2 Visualitzar calendari' with an 'EPIC' tag. It also shows 'Created by Marc Badia Romero' and the date '02 Jul 2019 10:45'. There are buttons for 'Link to Epic' and 'Add tag'. The description of the user story is 'Com a usuari, vull poder veure les properes partides per tal de poder veure qui i quan es juguen'. Below the description is a 'Subtasks' section with a green plus button to add more. The subtasks listed are: '#6 Filtrar per joc', '#7 Filtrar per lliga', '#8 Filtrar per equip', and '#10 Amagar spoilers tipus playoff'. Each subtask has a 'New' status with a dropdown arrow and a 'Not assign...' status with a star icon.

Subtasks		
#6 Filtrar per joc	New ▾	Not assign...
#7 Filtrar per lliga	New ▾	Not assign...
#8 Filtrar per equip	New ▾	Not assign...
#10 Amagar spoilers tipus playoff	New ▾	Not assign...

Figura 3. Exemple de tasca en Taiga
Font: Captura de pantalla de <https://taiga.io/>

A més, s'ha creat un document en Excel on es posa el temps dedicat cada dia al projecte, amb una petita descripció del que s'ha fet. Aquest Excel i tota la documentació del projecte (com per exemple aquest document) es guarda en *Google Drive* per poder-hi accedir fàcilment.

Finalment, la comunicació amb la directora del projecte es du a terme mitjançant correu electrònic.

4.3. Mètodes de validació

- **Validació de tasques**

Com s'ha explicat anteriorment, per a cada tasca s'han implementat una sèrie de tests automàtics que han garantit que la funcionalitat sigui l'esperada. No s'ha passat a la següent tasca fins que no s'ha comprovat que l'anterior funciona sense problemes.

- **Validació de sprint**

Al final de cada *sprint* s'ha fet una reunió amb la directora del projecte per rebre feedback del treball fet fins al moment. A més, en aquestes reunions s'ha modificat la planificació del següent *sprint* si és necessari i s'ha intentat millorar el mètode de treball.

- **Validació final**

Per tal de finalitzar el projecte s'ha fet una prova manual de totes les funcionalitats de l'aplicació en casos extrems.

- **Test d'usabilitat amb altres usuaris**

S'han fet diversos tests d'usabilitat durant el transcurs del projecte amb usuaris amb diferents nivells de coneixement de *e-sports* i d'informàtica en general. Els resultats d'aquests tests han servit per millorar l'aplicació i fer que sigui més accessible a tots els públics.

5. Planificació temporal

El projecte va començar oficialment el dia 9/7/2019 i acaba amb la defensa del projecte, el dimarts 28/01/2019.

Es va decidir treballar una mitja de 12 hores setmanals les setmanes de vacances (estiu i nadal) i una mitja de 20 hores setmanals les altres setmanes, el que comporta un total de 532 hores de feina previstes.

En aquest capítol es descriuen i estimen les tasques del projecte (secció 5.1 i 5.2), es mostra el diagrama de Gantt de la planificació temporal (secció 5.3) i s'expliquen els plans alternatius als possibles obstacles del projecte (secció 5.4).

5.1. Descripció i estimació de les tasques

Totes les tasques han estat realitzades per l'autor del projecte amb un ordinador de sobretaula i/o un portàtil ASUS.

5.1.1. Definició d'objectius i requisits

En aquesta primera tasca del projecte s'han definit els diferents objectius i requisits del projecte, el que ha facilitat les següents tasques considerablement. Més concretament, s'han definit les parts interessades que pot tenir el projecte i els objectius, subobjectius i requisits funcionals i no funcionals del sistema. Com és la primera tasca, no té dependències temporals.

Aquesta és la tasca més bàsica del projecte, on s'ha definit què farà l'aplicació. Per aquest motiu, i per la necessitat d'estudiar altres aplicacions similars ja existents (estat de l'art) i valorar quines respostes dels usuaris s'afegeixen a l'abast del projecte i quines no, es va estimar que aquesta tasca comportaria 45 hores.

5.1.2. Tria i estudi de l'API

Una vegada definits els objectius i requisits, cal triar una API que s'adapti bé al projecte i sigui capaç de satisfer aquests objectius.

Per a completar aquesta tasca, s'han estudiat les diferents APIs disponibles al mercat, tenint en compte les dades que ofereixen, el preu i el nombre de connexions que permeten. Una vegada triada, s'han definit quines són les dades que caldrà buscar per altres mètodes perquè l'API no les dona.

Es va estimar que es trigarien 25 hores a fer aquest estudi, ja que calia considerar totes les opcions disponibles, i, en alguns casos, posar-se en contacte amb els creadors de l'API per demanar més informació.

5.1.3. Disseny de la BD

Com que existeix la possibilitat que l'API no retorni totes les dades necessàries per al correcte funcionament del projecte, ha estat necessari fer una base de dades pròpia on estiguin les dades rellevants de l'API i altres dades que es buscaran per altres mètodes.

Una vegada es saben els objectius i requisits del projecte, es pot començar amb aquesta tasca, perquè es disposarà de la informació de quines seran les dades necessàries. Primer, s'han estudiat els diferents tipus de bases de dades (relacionals i no relacionals) i s'ha arribat a la conclusió de quina és la millor per a aquest projecte. Després s'ha fet el disseny de la base de dades procurant que sigui el més eficient possible tenint en compte les crides que s'havien de fer més endavant.

Com la meua experiència dissenyant bases de dades era molt reduïda, primer era necessari estudiar i repassar les bases del disseny de bases de dades i fer algun exercici més fàcil per tal de practicar. També s'esperava haver de demanar consell a la directora del projecte. Per aquests motius el temps d'aquesta tasca augmenta fins a les 35 hores aproximades.

5.1.4. Integració API a BD

Aquesta tasca va començar quan es va triar l'API i dissenyar la BD. Com el mateix nom indica, es va integrar l'API a la base de dades amb les dades que es consideressin rellevants per al projecte. Per la natura de constant canvi i afegiment de dades d'aquest projecte, es volia fer que la base de dades s'actualitzés de manera automàtica cada cert nombre d'hores amb les dades més noves i actualitzades de l'API. Aquesta integració i actualització s'han dut a terme amb un script Python.

Es va estimar que aquesta part del projecte comportaria 50 hores. Realment, el fet d'integrar l'API comporta menys temps, però com s'ha d'automatitzar el procés, calia estudiar tots els casos límits i implementar solucions per evitar errors inesperats o, fins i tot, implementar un sistema de notificacions pels problemes que només es puguin arreglar de manera manual.

5.1.5. Disseny de l'aplicació Android

Aquesta tasca es pot fer una vegada estan definits els objectius i requisits, ja que no depèn del back-end. En aquesta tasca, s'han dissenyat els mock-ups de l'aplicació i s'han fet tests d'usabilitat entre diferents usuaris. També s'han fet el diagrama de classes i els diagrames de seqüència per les funcionalitats més importants. Tot això es va estimar que comportaria 35 hores.

5.1.6. Desenvolupament de la REST API

Una vegada estigui el disseny de la BD, es pot començar a implementar la REST API (tot i que no es podrà provar fins que la BD tingui les dades). En aquesta tasca simplement s'han definit i implementat en Python les crides necessàries perquè l'aplicació es pugui comunicar amb la base de dades, el que es va estimar que comportaria aproximadament 25 hores.

5.1.7. Desenvolupament del front-end de l'aplicació Android

Es pot dir que aquesta és una de les tasques més importants del projecte perquè és el que veurà l'usuari. Es va estimar que comportaria unes 50 hores de feina, ja que s'havien de fer tots els layouts i les funcionalitats de l'aplicació d'acord amb el disseny fet. L'aplicació s'ha fet en Android Studio

5.1.8. Fer les crides a la REST API

Aquesta tasca és una tasca complementària de l'anterior, pel que s'ha fet a la vegada. En aquesta tasca s'han fet les crides necessàries a la REST API, per tal de poder mostrar la informació necessària de la funcionalitat que s'estigués desenvolupant en aquell moment, fet que comportaria 25 hores de feina estimades. S'ha fet alhora que es desenvolupa la REST API, per tal no fer més crides de les necessàries i, com en la tasca anterior, s'ha fet en Android Studio.

5.1.9. Testing de l'aplicació

S'han fet tests de les funcionalitats de l'aplicació en *Android Studio* per garantir el correcte funcionament en tots els casos i s'han provat les funcionalitats amb un mòbil *Samsung S8+*. Es va estimar que comportaria 45 hores aproximadament i s'ha fet a la vegada que el desenvolupament del front-end (s'ha fet el test necessari després d'acabar cada funcionalitat).

5.1.10. Desenvolupament de serveis webs per al back-end

Una vegada la base de dades estigui operativa, es pot començar amb el desenvolupament de serveis webs per al back-end. Aquests serveis webs permetrien tenir més control sobre l'automatització de la base de dades i arreglar alguns dels errors que no es puguin arreglar de forma automàtica. L'estimació d'aquesta tasca era de 40 hores, però és altament variable en base a quins serveis es consideren finalment necessaris.

5.1.11. Gestió del projecte

Aquesta tasca no té cap dependència temporal. Comporta tot el relacionat amb la gestió del projecte i la documentació, i també està formada per diverses tasques més petites.

- **Reunions amb la directora.** S'han fet diverses reunions inicialment per tal de concretar els objectius i requisits, i també s'han fet cada vegada que s'acabi un sprint. Contant el temps de viatge surt a unes 40 hores.
- **Document de definició de l'abast i contextualització.** En la tasca de definició d'objectius i requisits ja s'haurà fet una part de la feina d'aquesta tasca, reduint així l'estimació en temps a 5 hores.
- **Document de la planificació temporal.** En aquest document s'han explicat les diferents tasques i la planificació temporal. Es va estimar que comportaria 10 hores.
- **Document de la gestió econòmica i sostenibilitat.** La redacció d'aquest document es va estimar que comportaria 10 hores, com l'anterior.
- **Document final i preparació de la defensa.** En aquesta tasca es redacta la memòria final del projecte i es prepara la defensa del TFG, el que es va estimar que comportaria unes 55 hores.

5.2. Taula resum de les tasques

Codi	Nom Descriptiu	Temps estimat (hores)	Dependències temporals	Recursos
T1	Definició d'objectius i requisits	45	-	Usuaris potencials, Google Form, ordinador, autor del projecte
T2	Tria i estudi de l'API	25	T1 < T2	Ordinador, autor del projecte
T3	Disseny de la BD	35	T1 < T3	Ordinador, PyCharm, GitHub, autor del projecte
T4	Integració API a BD	50	(T2, T3) < T4	Ordinador, PyCharm, GitHub, autor del projecte
T5	Afegir altra informació a BD	50	(T2, T3) < T5	Ordinador, PyCharm, GitHub, autor del projecte
T6	Disseny de l'aplicació Android	35	T1 < T6	Ordinador, autor del projecte
T7	Desenvolupament de REST API	25	T3 < T7	Ordinador, PyCharm, GitHub, Heroku, autor del projecte
T8	Desenvolupament del front-end de l'aplicació Android	50	(T3, T4, T5) < T7	Ordinador, Android Studio, GitHub, autor del projecte
T9	Fer les crides a la REST API	25	(T3, T4, T5) < T7	Ordinador, GitHub, autor del projecte
T10	Testing de l'aplicació	45	(T3, T4, T5) < T7	Ordinador, Google Drive, autor del projecte
T11	Desenvolupament de serveis webs per al back-end	40	(T4, T5) < T8	Ordinador, GitHub, autor del projecte
T12	Gestió del projecte	120	-	Ordinador, Google Drive, autor del projecte
TOTAL	-	545	-	-

Taula 1. Resum de les tasques

Font: Elaboració pròpia

NOTA: La tasca “Afegir altra informació a BD” s’explica al apartat de Gestió del risc, i s’ha afegit a la planificació temporal i al diagrama de Gantt per l’alta probabilitat que té aquest risc.

5.3. Diagrama de Gantt

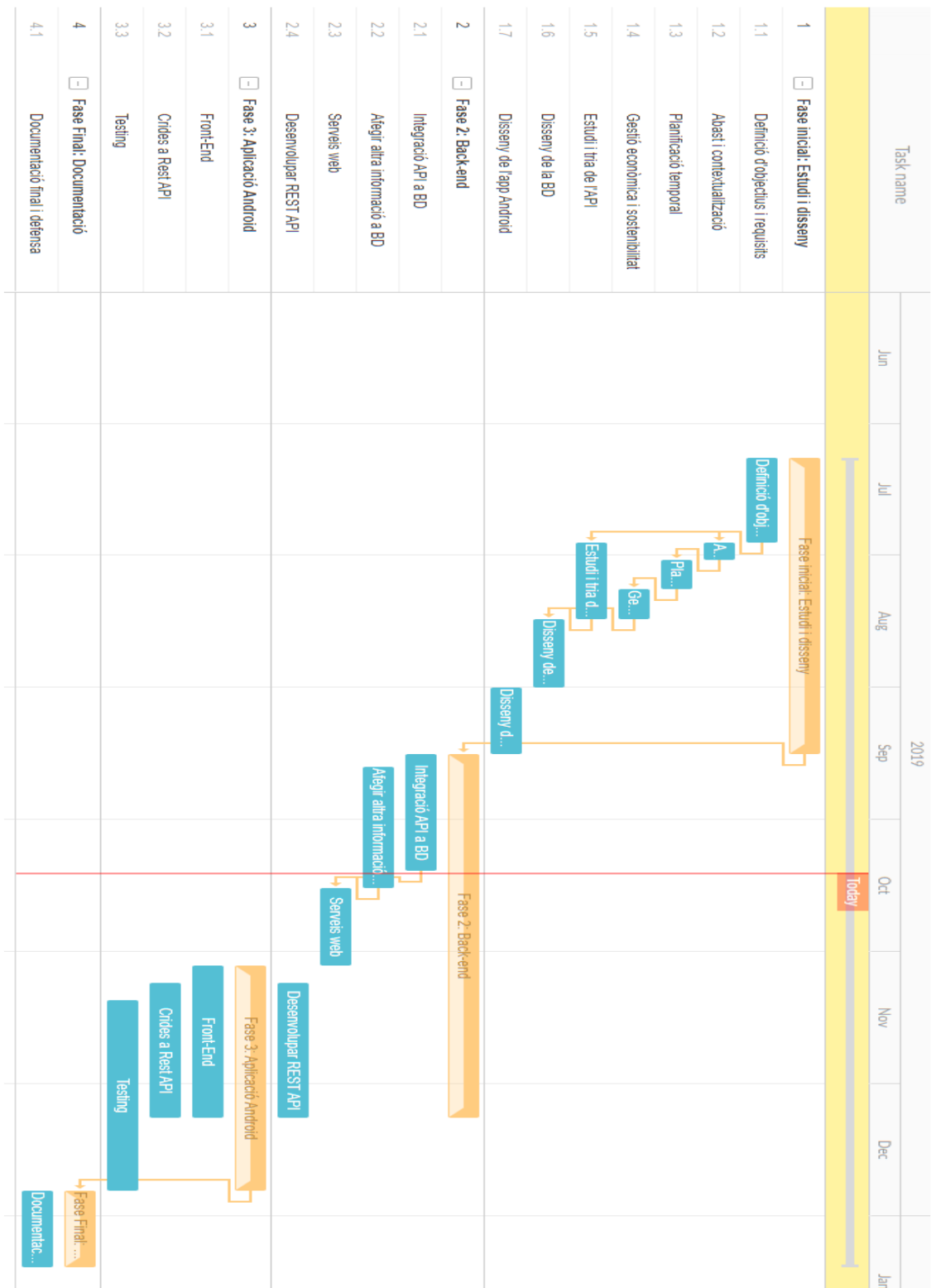


Figura 4. Diagrama de Gantt del projecte
Font: Elaboració pròpia utilitzant ganttpro.com

5.4. Gestió del risc: Plans alternatius i obstacles

5.4.1. Reducció de l'abast

Si per qualsevol motiu es veia que no s'arribaria a l'abast establert, es preguntaria als stakeholders quina tasca creien que era la menys prioritària per no fer-la. En tot cas, i com es pot comprovar en el diagrama de Gantt, es van deixar dues setmanes de marge entre la finalització del projecte i el dia de la defensa.

5.4.2. Limitacions de l'API

Era molt probable que l'API triada no fos perfecte per al projecte i faltés informació important o rellevant. En aquest cas s'afegiria una nova tasca anomenada "Afegir altra informació a BD":

Un cop definides quines són les dades extra que es necessiten a la base de dades, començaria aquesta tasca. Per a cada dada que fos necessària, s'estudiaria d'on es podria extreure i quina era la manera més òptima per a fer-ho per tal que fos un procés automatitzat, és a dir, que no calgués afegir aquestes dades manualment. Com en la tasca anterior, les dades s'afegirien mitjançant un script de *Python*.

Com la tasca d'integrar l'API a BD, aquesta tasca també es va estimar que comportaria 50 hores pels mateixos motius. Es faria per l'autor del projecte amb un ordinador amb *Pycharm* i es pujaria el codi al repositori del back-end a *GitHub*. Cal dir que la planificació temporal s'ha fet sabent que aquest risc es compliria i caldria fer aquesta nova tasca.

5.4.3. Legalitat de les dades

Llegint sobre la legalitat de mostrar dades d'equips o jugadors en aplicacions de tercers, es va entendre que depenia de la mateixa empresa decidir qui pot utilitzar-les i qui no pot. Per aquest motiu, es contactaria amb l'empresa per demanar permís d'utilitzar les dades. En cas que es denegés el permís, caldria una nova tasca estimada en 15 hores, en la que s'estudiarien els requisits legals per tal de poder llançar l'aplicació al mercat. A diferència del risc anterior, aquest no s'ha considerat en la planificació temporal, ja que la probabilitat que succeís és molt petita. En el pitjor cas, l'aplicació es quedaria com a TFG i no es llençaria al mercat.

5.4.4. Disponibilitat API

Era possible que en algun moment l'API deixés de funcionar durant un temps determinat. En aquest cas no es podia fer molt, ja que durant aquest període de temps no es podrien afegir dades noves, però com que la base de dades guarda les dades antigues, es podria treballar amb aquestes, i l'aplicació seguiria funcionant encara que no es tinguessin les dades més recents.

6. Gestió econòmica

En aquest capítol es presenta la gestió econòmica feta a l'inici del projecte. Primer s'identifiquen i s'estimen els diferents costos del projecte (secció 6.1) i després es defineixen els indicadors usats per tal de controlar les desviacions del pressupost (secció 6.2).

6.1. Identificació i estimació dels costos

En aquest apartat es fa una estimació del cost total del projecte, justificant primer els costos dels recursos humans dividits per rol (tot i que en aquest projecte només hi treballa una persona) i després els recursos de hardware i software, a més d'altres despeses. Al final de l'apartat s'inclou una taula resumint tots els costos.

6.1.1. Recursos humans

Aquest projecte s'ha desenvolupat per una sola persona, que ha fet tots els rols propis d'un projecte de desenvolupament de software. Per poder estimar correctament els costos relacionats als recursos humans, primer és necessari assignar un preu per hora a cada rol del projecte. Aquesta informació que es mostra en la taula 2 s'ha extret d'un article de la consultoria *PageGroup* on es detalla el salari mínim i màxim d'aquests rols a Espanya [16].

Rol	Descripció	Cost per hora (€)
Cap de projecte	Gestiona el projecte i/o l'equip que forma el projecte	24
Analista	Troba els objectius i requisits del projecte	18
Arquitecte	Dissenya el projecte	18
Programador Front-end	Desenvolupa la part de l'aplicació Android del projecte	16
Programador Back-end	Desenvolupa la part de base de dades, REST API i serveis web del projecte	16
Tester	Comprova, realitzant diverses proves, que el codi generat per els programadors funcioni correctament	15

Taula 2. Cost per hora dels recursos humans
Font: Elaboració pròpia

Seguidament, com es veu a la taula 3, s'assignen les hores de feina que cada rol li dedica a cada tasca i es calcula el cost per tasca:

Tasca	Hores per rol						Cost (€)
	Cap de projecte	Analista	Arquitecte	Front-end	Back-end	Tester	
Definició d'objectius i requisits	10	35					870,00
Tria i estudi de l'API	5	20					480,00
Disseny de la BD	5		30				660,00
Disseny de l'aplicació Android	5		30				660,00
Integració API a BD					50		800,00
Desenvolupament de la REST API					25		400,00
Desenvolupament del front-end de l'aplicació Android				50			800,00
Fer les crides a la REST API				25			400,00
Testing de l'aplicació						45	675,00
Desenvolupament de serveis webs per al back-end					40		640,00
Reunions amb la directora	40						960,00
Document de definició de l'abast i contextualització	5						120,00
Document de la planificació temporal	10						240,00
Document de la gestió econòmica i sostenibilitat	10						240,00
Document final i preparació de la defensa	55						1320,00
TOTAL	145	55	60	75	115	45	9265,00

*Taula 3. Cost per tasca
Font: Elaboració pròpia*

6.1.2. Costos genèrics

En aquest apartat s'han calculat els costos dels recursos hardware i software esmentats amb anterioritat i altres despeses necessàries per dur a terme el projecte. Pel que fa als recursos hardware i software, només s'han considerat recursos amb cost, pel que no es mostren els recursos software gratuïts. Dit això, es mostra el preu real dels recursos software amb llicència gratuïta a estudiants, per tal de simular correctament el preu real del projecte.

Per tal d'establir un cost d'amortització, s'ha assumit que un any de vida útil equival a 260 dies de feina amb una mitja de 3,5 hores al dia. Per tant, el cost d'amortització serà $CA = \text{Hores d'ús} * (\text{Preu recurs} / (X \text{ anys de vida útil} * 260 \text{ dies/any} * 3,5 \text{ hores/dia}))$.

Les hores que es mostren en la taula 4 són aproximacions de les hores d'ús dels recursos. La majoria d'hores del projecte s'han fet amb l'ordinador de sobretaula, mentre que el portàtil s'ha utilitzat per coses puntuals i per les reunions amb la directora. El telèfon mòbil s'ha utilitzat per tal de provar l'aplicació.

Recurs	Cost Total (€)	Vida útil (anys)	Hores d'ús	Cost d'amortització (€)
Ordinador de sobretaula	1200	5	390	102,86
Portàtil HP	599	4	60	9,87
Mòbil Samsung Galaxy S8 +	799,9	4	30	6,59
JetBrains Product Pack (inclou PyCharm)	649	1	140	99,85
TOTAL	3247,9	-	-	219,17

Taula 4. Preu i cost d'amortització dels recursos hardware i software

Font: Elaboració pròpia

En la següent taula es descriuen altres despeses. Algunes coses a tenir en compte són que el consum energètic és el total del consum dels dos ordinadors i el telèfon mòbil; el desplaçament es refereix al cost de desplaçar-se a la facultat amb transport públic i els costos d'espai inclouen la meua part del lloguer del meu habitatge.

Recurs	Cost Total	Quantitat	Cost (€)
Consum energètic	0,16 €/kWh	2 kWh per dia. 150 dies aproximats de feina	48
Internet	63 €/mes	390 hores	34,12
Desplaçament	112,40€	12 desplaçaments	24
Costos d'espai	620 €/mes	3 persones	206,67
TOTAL	-	-	312,79

Taula 5. Altres despeses
Font: Elaboració pròpia

6.1.3. Imprevistos i pla de contingència

En el transcurs del projecte podien haver-hi uns imprevistos que poden afectar el pressupost, pel que es va establir una ràtio de contingència el 15%. Aquests imprevistos són:

- **Avaria d'ordinador.** Existia la possibilitat que l'ordinador de sobretaula on es realitza la gran majoria del treball deixés de funcionar, pel que s'hauria de reparar.
- **Limitacions de l'API.** Podia ser que l'API trobada no assolís tots els requisits. Si això passés, es va estimar que la nova tasca requeriria 50 hores per al programador de back-end, amb un cost total de 800 €.
- **Pèrdua de dades.** En aquest projecte s'han realitzat còpies de seguretat diàriament. No obstant això, era possible que així i tot es perdés com a màxim un dia de feina per motius externs, com una fallada del repositori de GitHub. El cost mitjà de recuperar aquest dia de feina seria de 32,50 €.
- **Legalitat de les dades.** Si es trobés que es denegues el permís a utilitzar certes dades, es faria una tasca de 15 hores on s'estudiarien els requisits per tal de poder-les utilitzar legalment. Això costaria 270 €.

Imprevist	Cost Total (€)	Percentatge de probabilitat (%)	Cost (€)
Avaria d'ordinador	200	10	20,00
Limitacions de l'API	800	80	640,00
Pèrdua de dades	32,5	5	1,63
Legalitat de les dades	270	5	13,50

Taula 6. Imprevistos
Font: Elaboració pròpia

6.1.4. Pressupost total

Activitat	Import (€)
Definició d'objectius i requisits	870,00
Tria i estudi de l'API	480,00
Disseny de la BD	660,00
Disseny de l'aplicació Android	660,00
Integració API a BD	800,00
Desenvolupament de la REST API	400,00
Desenvolupament de l'aplicació Android (Front-end)	800,00
Fer les crides a la REST API	400,00
Testing	675,00
Desenvolupament de serveis webs per al back-end	640,00
Reunions amb la directora	960,00
Document de definició de l'abast i contextualització	120,00
Document de la planificació temporal	240,00
Document de la gestió econòmica i sostenibilitat	240,00
Document final i preparació de la defensa	1320,00
TOTAL CPA	9265,00
Amortització recursos hardware	104,81
Amortització recursos software	78,45
Consum energètic	48,00
Internet	34,12
Desplaçament	24,00
Costos d'espai	206,67
TOTAL CG	496,05
TOTAL Costos (Total CPA + Total CG)	9761,05
Contingència	1464,16
TOTAL CD + CI + Contingència	11225,21
Avaria d'ordinador	200,00
Limitacions de l'API	640,00
Pèrdua de dades	1,63
Legalitat de les dades	13,50
TOTAL Imprevistos	855,13
TOTAL	12080,34

Taula 7. Pressupost del projecte

Font: Elaboració pròpia

6.2. Control de gestió

Per tal de poder controlar el pressupost, es va decidir calcular alguns indicadors que diguin si el pressupost del projecte s'està desviant molt. Aquests indicadors s'havien de calcular en finalitzar cada tasca i s'han comparat amb l'estimació del pressupost feta anteriorment, per tal de poder corregir-la en cas que fos necessari. Cal dir que aquests indicadors no s'han calculat pels recursos de hardware ni els de software, ja que els preus són els de venda al públic.

Pel que fa als recursos humans, s'han registrat les hores reals dedicades a cada tasca per part de cada rol, amb el fi de poder veure les possibles desviacions temporals i poder modificar el pressupost si escau.

Una vegada es tenen les hores reals de feina (HR) es calcula el cost real (CR) i es comparen amb les hores estimades (HE) i el cost Estimat per a cada recurs humà i per a cada tasca.

D'aquí treiem tres desviacions per a cada recurs humà per tasca (també s'han calculat per les tasques dels imprevistos si escau):

- **Desviament en preu:** $DP = (CE - CR) * HR$
- **Desviament en consum:** $DC = (HE - HR) * CE$
- **Desviament total:** $DT = DP + DC$

Amb aquests indicadors podem saber amb exactitud on s'han produït les desviacions importants del projecte, quin recurs humà les ha produït i la quantitat o import de la desviació.

Quan es detectés una desviació mitjanament important (superior al 0,5% del pressupost total del projecte), s'ha estudiat la raó de la desviació (mala planificació, imprevist, subestimació de la tasca...) per tal d'evitar cometre el mateix error en tasques futures.

Pel que fa als costos genèrics i als imprevistos que no requereixin desenvolupar tasques addicionals, s'han calculat els següents indicadors:

- **Desviament total d'un cost fixe:** total cost fixe pressupostat - total cost fixe real.
- **Desviament de l'amortització d'un cost:** total amortització pressupostada - total amortització real.

En aquest cas també s'han estudiat les possibles causes dels desviaments per tal de millorar la planificació en el futur.

7. Especificació de requisits

En aquest capítol es detallen els requisits del sistema a desenvolupar, així com el procés seguit per a l'obtenció dels requisits (secció 7.1). Els requisits funcionals (secció 7.2) es descriuen mitjançant un diagrama de casos d'ús, i la descripció detallada de cada cas d'ús. S'inclou un model conceptual per ajudar en la comprensió dels requisits que descriu el context dels jocs online dels que es vol donar informació a través de l'aplicació a desenvolupar (secció 7.3). Finalment, els requisits no funcionals es descriuen en la secció 7.3, amb el criteri d'acceptació de cadascun d'ells.

7.1. Procés d'obtenció de requisits

Aquest projecte ha estat ideat i desenvolupat per una sola persona, per tant els seus requisits no venien definits per cap empresa.

Per tal d'obtenir aquests requisits s'han utilitzat diversos mètodes. Primer, es va fer un *brainstorming* d'on van sortir els requisits bàsics del sistema, d'acord amb el que jo creia que era necessari en una aplicació d'aquest estil com a possible usuari.

Altres requisits es van obtenir estudiant pàgines web i aplicacions de dispositius mòbils similars (i no tan similars) a la idea del sistema del projecte, les vistes a l'estat de l'art i altres no relacionades amb els *e-sports*, com poden ser aplicacions d'esports tradicionals o fòrums d'Internet. Alguns d'aquests requisits es van mantenir gairebé intactes, mentre que d'altres només es va mantenir la idea original, amb una funcionalitat completament diferent.

També es va preguntar per l'opinió de gent. Es va preguntar personalment a un total de 15 persones interessades en una aplicació que tingués la idea d'aquest projecte per què esperarien, voldrien i no voldrien d'una aplicació d'estadístiques i resultats de competicions d'*e-sports*. Moltes de les idees donades ja s'havien tingut en compte en les fases anteriors, però algunes van resultar de molt útils per avançar amb l'obtenció de requisits, sobretot les parts que no voldrien veure en una aplicació d'aquest estil. Per exemple, 5 de les persones enquestades opinaven que no era necessari tenir més de 5 estadístiques per jugador, ja que podria resultar aclaparador, pel que es va modificar aquest requisit, que en un principi deia que es mostrarien un total de 7 estadístiques per jugador.

Una vegada es va recopilar una llista de requisits tant funcionals com no funcionals, es van eliminar els que no eren viables o s'allunyaven molt de la idea original del projecte amb l'ajuda de la directora del projecte, com pot ser la possibilitat de veure partides en diferit a través de l'API de Youtube. Aquest requisit va ser dels primers, però es va acabar desestimant perquè no es va trobar cap manera viable d'aconseguir de forma automàtica tots els enllaços als vídeos de les partides. Alguns d'aquests requisits eliminats s'han quedat com a possible treball futur si es troba una manera viable de fer-los.

Una vegada es tenia la llista completa de tots els requisits, es van concretar els funcionals de manera que tots tinguessin un escenari principal d'èxit i les extensions necessàries. Pel que fa als no funcionals, es va definir un criteri d'acceptació per a cadascun. Després es va preguntar per l'opinió de la directora i d'algunes de les persones enquestades anteriorment dels requisits més importants, i es van modificar d'acord amb les respostes donades.

7.2 Context

Abans de posar els requisits cal descriure el context de les competicions online de videojocs.

En aquest sistema s'incorporen estadístiques i dades de dos *e-sports*: *League of Legends (LoL)* i *Counter Strike: Global Offensive (CSGO)*. Un *e-sport* no és més que un terme abstracte per referir-se a un videojoc del qual es fan competicions a nivell professional, és a dir, els jugadors i les organitzacions reben diners per competir.

Tant *LoL* com *CSGO* són videojocs en els quals, en cada partida, s'enfronten dos equips de cinc jugadors cadascun. En les competicions professionals, existeix el concepte de *sèrie*, que és un enfrontament competitiu entre dos equips. Aquestes sèries tenen molts formats depenent de la competició, però normalment segueixen l'estructura del *Best-of*, on els dos equips juguen un conjunt de partides. Per exemple, en un *Best-of 1*, els equips jugarien una única partida, i el guanyador d'aquesta partida seria el guanyador de la sèrie. En un *Best-of 5*, els equips jugarien un màxim de 5 partides, i el guanyador de la sèrie serà l'equip que aconsegueixi guanyar la majoria de les partides, és a dir, el primer que en guanyi tres. Per posar un exemple d'un esport tradicional com el basketball, el format *Best-of 7* s'utilitza en els Play-Offs de la NBA, on un equip passa de ronda (o el que és el mateix, guanya la sèrie) quan guanya quatre partits. Per tant una sèrie és un grup o conjunt de partides entre dos equips.

Per norma general, un jugador només forma part d'un equip en un moment del temps determinat. Hi ha certes excepcions com el cas de les cessions, on un jugador juga en un equip tot i formar part d'un altre, però aquests casos excepcionals no s'han tingut en compte en l'àmbit d'aquest sistema; un jugador forma part de l'equip en el qual juga en aquell moment del temps. Per altra banda, un equip, i per tant, un jugador, pot formar part de més d'una competició a la vegada

7.3. Requisits funcionals

Primer es detallen els requisits funcionals, a través de casos d'ús.

7.3.1. Diagrames de casos d'ús

Aquí es mostren els diferents diagrames dels casos d'ús, per tal de poder veure una representació gràfica de les relacions entre els actors del sistema.

Primer, com es mostra en la figura 5 i per tal de simplificar conceptes, es considera que un usuari que ha iniciat sessió en el sistema és un “Usuari registrat”, i un usuari registrat amb permisos de moderador és un “Moderador”. Per tant, un moderador pot invocar els casos d’ús d’un usuari, però un usuari no pot invocar els casos d’ús d’un moderador.

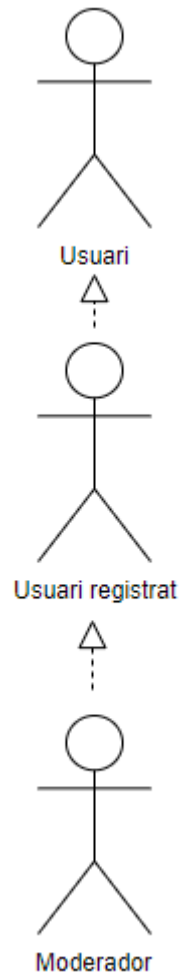


Figura 5. Especialització dels usuaris
Font: Elaboració pròpia

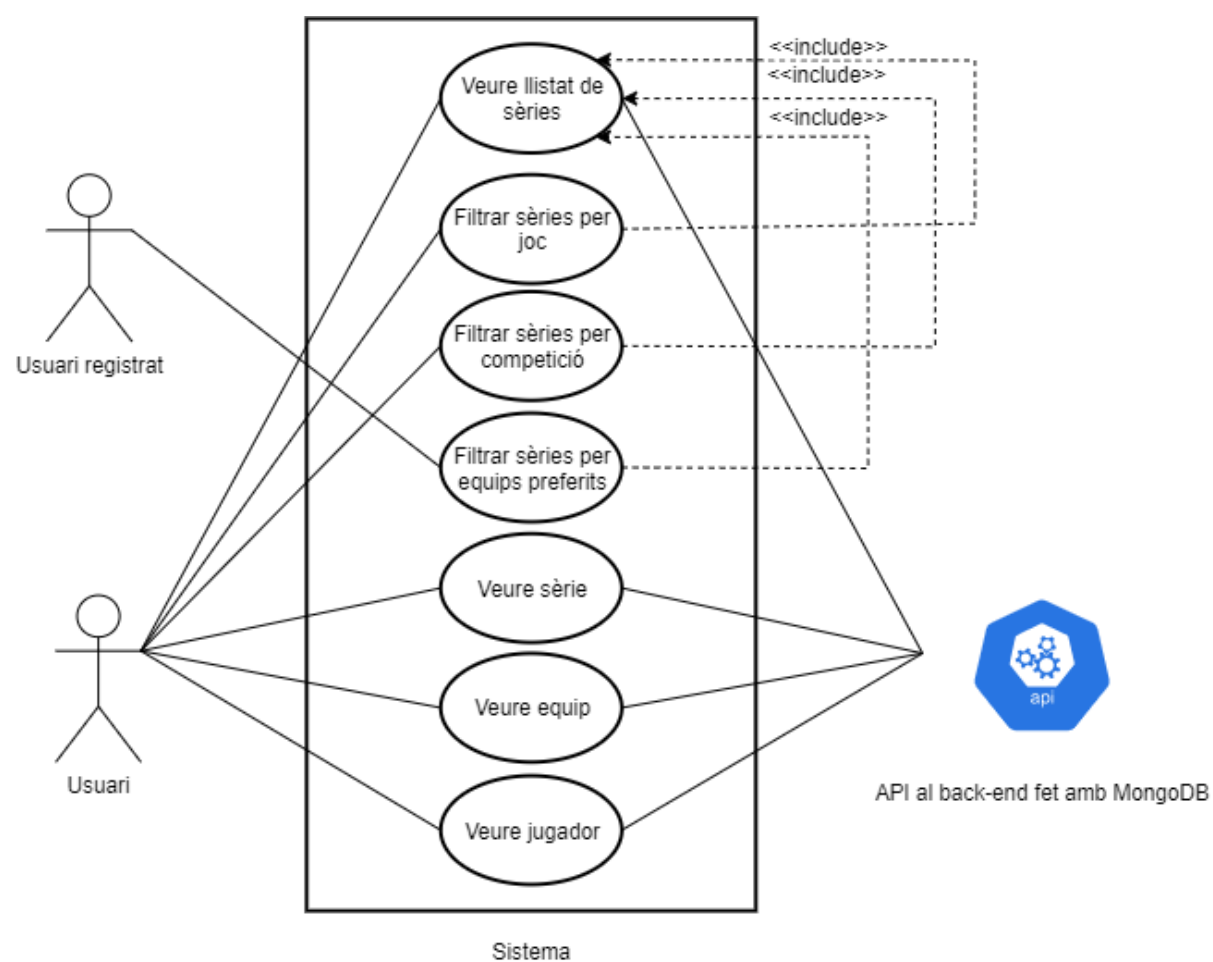


Figura 6. Diagrama dels casos d'ús relacionats amb visualització de dades
 Font: Elaboració pròpia

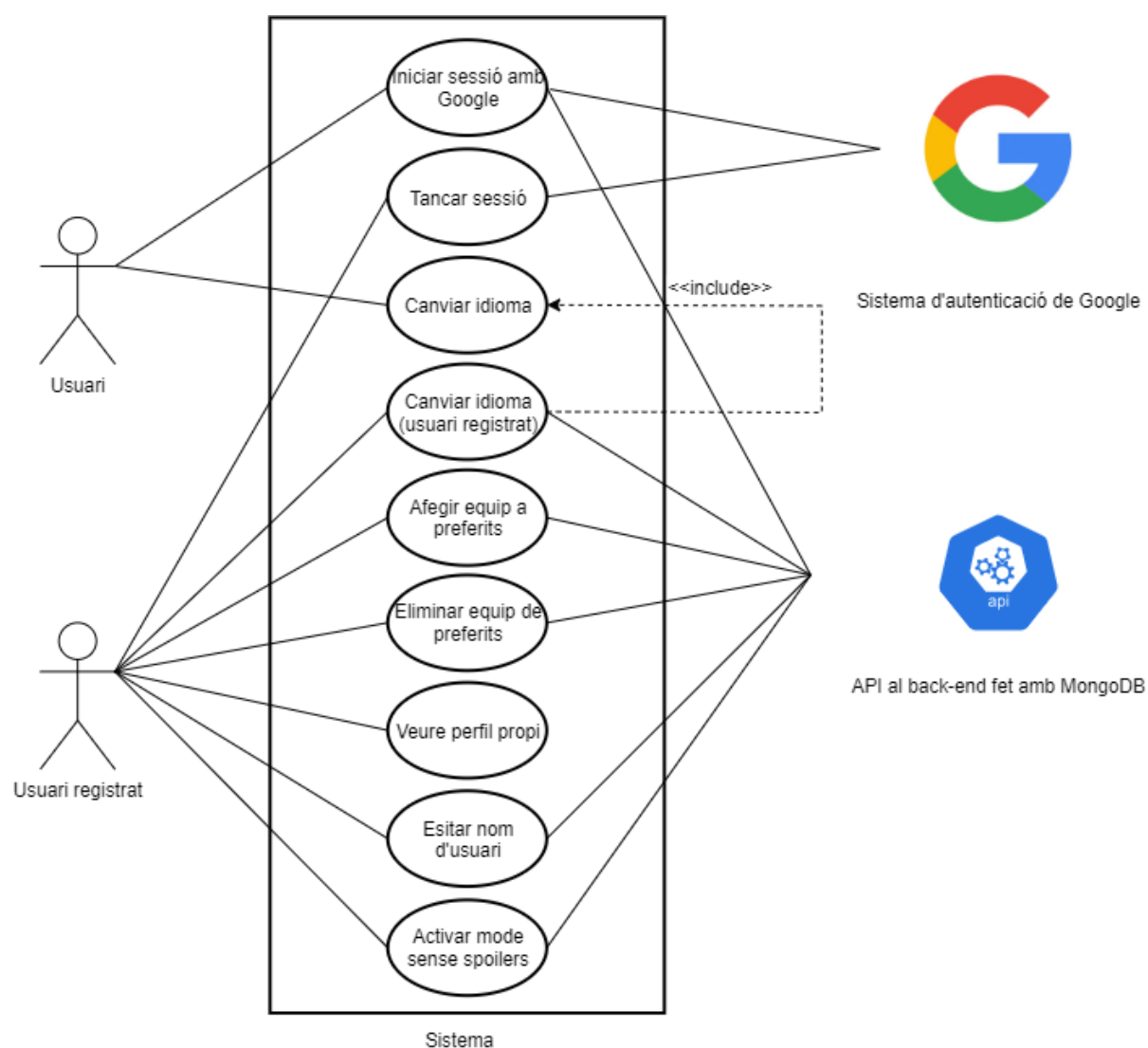


Figura 7. Diagrama dels casos d'ús relacionats amb la sessió
Font: Elaboració pròpia

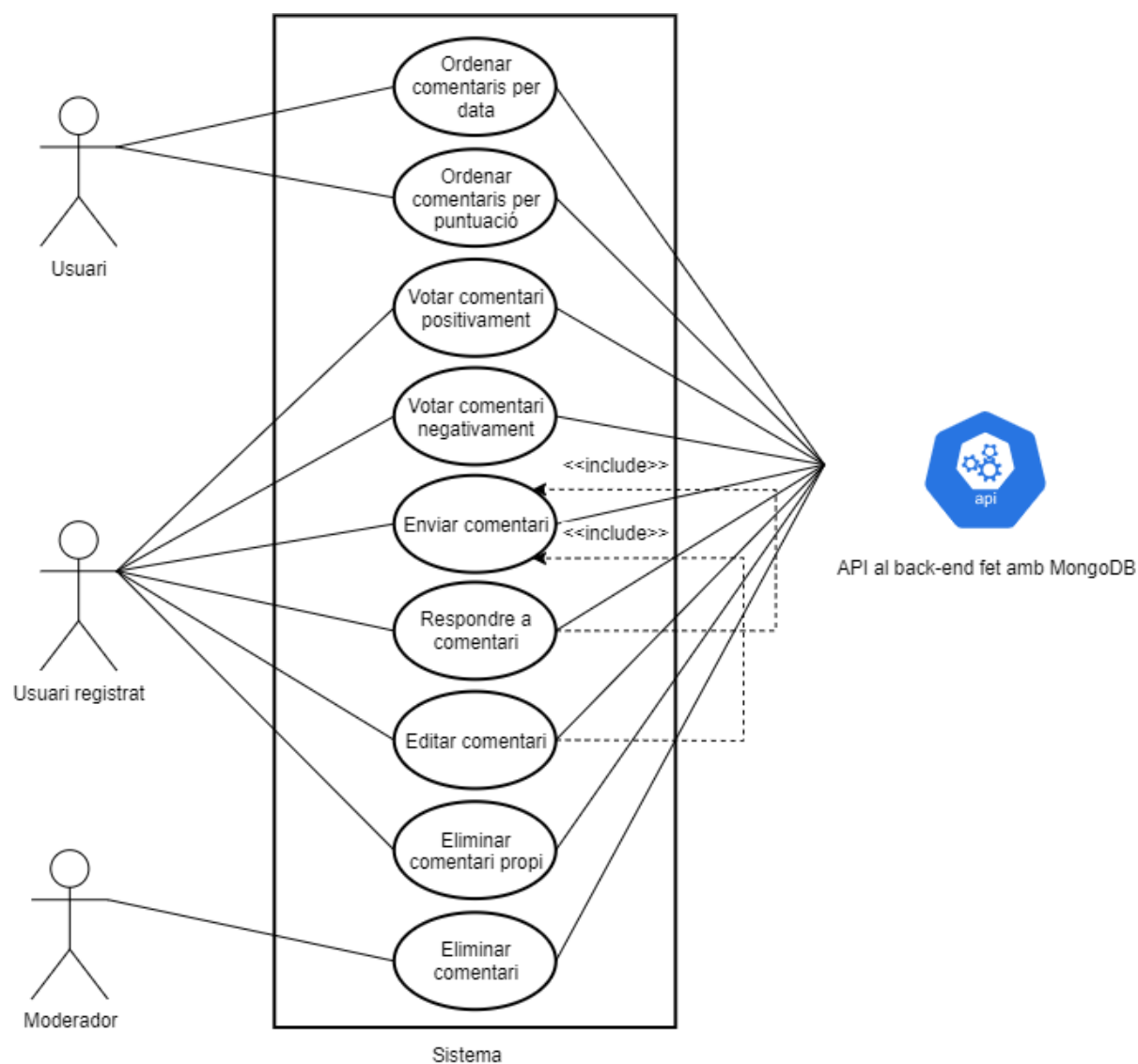


Figura 8. Diagrama dels casos d'ús relacionats amb els comentaris
Font: Elaboració pròpia

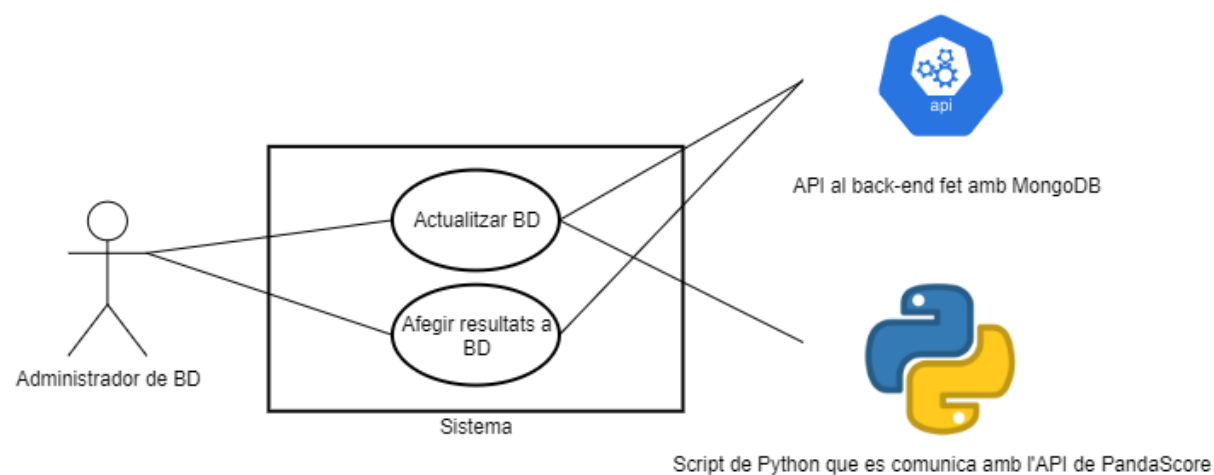


Figura 9. Diagrama dels casos d'ús relacionats amb el manteniment de la BD
Font: Elaboració pròpia

7.3.2. Descripció de casos d'ús

En aquest apartat es descriuen els casos d'ús mostrats en els diagrames anteriors. Cada cas d'ús disposa d'una taula on s'especifica el nom del cas d'ús, l'actor principal que el du a terme, les precondicions que s'han de complir, el disparador (l'acció que posa en marxa el cas d'ús), l'escenari principal d'èxit on es descriu el funcionament del cas d'ús si s'executa correctament i les extensions, què indiquen les accions que succeeixen en el cas que l'escenari principal d'èxit no s'executi correctament. Totes les taules són d'elaboració pròpia.

Cas d'ús	#1 Veure llistat de sèries	Actor principal	Usuari
Precondicions	-		
Disparador	L'usuari ha iniciat l'aplicació o vol veure el llistat de les sèries		
Escenari principal d'èxit			
<div>1. L'usuari inicia l'aplicació o indica al sistema que vol veure el llistat de les sèries.</div> <div>2. El sistema es comunica amb l'API, que retorna una llista amb les properes sèries i una llista amb les sèries acabades.</div> <div>3. El sistema guarda les sèries rebudes.</div> <div>4. El sistema redirigeix a l'usuari a la vista de "Llista de sèries", on es mostren les properes sèries en una pestanya i les acabades en una altra.</div>			
Extensions			
<div>2a. Alguna de les llistes retornades és buida.</div> <div>2a1. El sistema indica a l'usuari que no s'ha trobat cap sèrie.</div>			

Cas d'ús	# 2 Filtrar sèries per joc	Actor principal	Usuari
Precondicions	-		
Disparador	L'usuari vol filtrar les sèries per un dels jocs		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol veure les sèries de només un dels jocs.</div> <div>2. El sistema es comunica amb l'API, que retorna una llista amb les properes sèries i una llista amb les sèries acabades demanades.</div> <div>3. El sistema guarda les sèries rebudes.</div> <div>4. El sistema redirigeix a l'usuari a la vista de "Llista de sèries ", on es mostren les properes sèries en una pestanya i les acabades en una altra.</div>			
Extensions			
<div>2a. Alguna de les llistes retornades és buida.</div> <div>2a1. El sistema indica a l'usuari que no s'ha trobat cap sèrie</div>			

Cas d'ús	#3 Filtrar sèries per competició	Actor principal	Usuari
Precondicions	-		
Disparador	L'usuari vol filtrar les sèries per alguna de les competicions		
Escenari principal d'èxit			

<ol style="list-style-type: none"> 1. L'usuari indica al sistema que vol veure les sèries de només una de les competicions. 2. El sistema es comunica amb l'API, que retorna una llista amb les properes sèries i una llista amb les sèries acabades demanades. 3. El sistema guarda les sèries rebudes. 4. El sistema redirigeix a l'usuari a la vista de "Llista de sèries", on es mostren les properes sèries en una pestanya i les acabades en una altra.
Extensions
2a. Alguna de les llistes retornades és buida. 2a1. El sistema indica a l'usuari que no s'ha trobat cap sèrie.

Cas d'ús	#4 Canviar idioma	Actor principal	Usuari
Precondicions	-		
Disparador	L'usuari vol canviar l'idioma de l'aplicació		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol canviar l'idioma de l'aplicació des de la vista de “Configuració”.</div> <div>2. El sistema canvia l'idioma de l'aplicació per la sessió actual.</div> <div>3. El sistema redirigeix a l'usuari a la vista principal.</div>			
Extensions			
-			

Cas d'ús	#5 Veure sèrie	Actor principal	Usuari
Precondicions	-		
Disparador	L'usuari vol veure la informació d'una sèrie		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol veure l'informació d'una sèrie en concret.</div> <div>2. El sistema es comunica amb l'API, que retorna la informació adient incloent els comentaris i la informació de les partides individuals de la sèrie.</div> <div>3. El sistema guarda aquestes dades.</div> <div>4. El sistema redirigeix l'usuari a la vista "Sèrie", on es mostra la informació en diferents pestanyes.</div>			
Extensions			
<div>4a. La sèrie no ha finalitzat.</div> <div>4a1. El sistema només mostra la pestanya de "Prèvia".</div>			

Cas d'ús	#6 Ordenar els comentaris per data	Actor principal	Usuari
Precondicions	Hi ha com a mínim un comentari.		
Disparador	L'usuari vol ordenar els comentaris per data.		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol ordenar un set de comentaris en concret (és a dir, els comentaris de la prèvia o el post-partit d'una sèrie en concret) per data.</div> <div>2. El sistema es comunica amb l'API, que retorna el llistat de comentaris ordenats per data.</div> <div>3. El sistema mostra els comentaris ordenats per data.</div>			

Extensions
-

Cas d'ús	#7 Ordenar els comentaris per puntuació	Actor principal	Usuari
Precondicions	Hi ha com a mínim un comentari.		
Disparador	L'usuari vol ordenar els comentaris per puntuació.		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol ordenar un set de comentaris en concret (és a dir, els comentaris de la prèvia o el post-partit d'una sèrie en concret) per puntuació.</div> <div>2. El sistema es comunica amb l'API, que retorna el llistat de comentaris ordenats per puntuació.</div> <div>3. El sistema mostra els comentaris ordenats per puntuació.</div>			
Extensions			
-			

Cas d'ús	#8 Veure equip	Actor principal	Usuari
Precondicions	-		
Disparador	L'usuari vol veure la informació d'un equip en concret		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol veure la informació d'un equip en concret.</div> <div>2. El sistema es comunica amb l'API, que retorna la informació de l'equip demanat.</div> <div>3. El sistema guarda la informació rebuda.</div> <div>4. El sistema redirigeix l'usuari a la vista “Equip”, on es mostra la informació del equip demanat.</div>			
Extensions			
-			

Cas d'ús	#8 Veure jugador	Actor principal	Usuari
Precondicions	-		
Disparador	L'usuari vol veure la informació d'un jugador en concret		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol veure la informació d'un jugador en concret.</div> <div>2. El sistema es comunica amb l'API, que retorna la informació del jugador demanat.</div> <div>3. El sistema guarda la informació rebuda.</div> <div>4. El sistema redirigeix l'usuari a la vista "jugador ", on es mostra la informació del jugador demanat.</div>			
Extensions			
-			

Cas d'ús	#9 Iniciar sessió amb Google	Actor principal	Usuari
-----------------	------------------------------	------------------------	--------

Precondicions	-
Disparador	L'usuari vol iniciar sessió en el sistema.
Escenari principal d'èxit	
<ol style="list-style-type: none"> 1. L'usuari indica al sistema que vol iniciar sessió amb Google. 2. El sistema es comunica amb el servei de Google, que retorna el ID del usuari demanat. 3. El sistema es comunica amb l'API, que retorna la informació del usuari demanat. 4. El sistema guarda aquesta informació. 5. El sistema crea una sessió per a aquest usuari. 6. El sistema redirigeix l'usuari a la vista "Perfil". 	
Extensions	
<ol style="list-style-type: none"> 3a. No s'ha trobat cap usuari amb el ID especificat. <ol style="list-style-type: none"> 3a1. El sistema es comunica amb el servei de Google, que retorna les dades bàsiques de l'usuari demanat. 3a2. El sistema es comunica amb l'API, que crea aquest nou usuari a la base de dades 3a3. Es torna al pas 3. 	

Cas d'ús	#10 Tancar sessió	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema.		
Disparador	L'usuari vol tancar sessió en el sistema.		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol tancar sessió.</div> <div>2. El sistema pregunta a l'usuari si realment vol tancar sessió.</div> <div>3. L'usuari respon que sí.</div> <div>4. El sistema elimina la instància de sessió.</div>			
Extensions			
<div>3a. L'usuari respon que no.</div> <div>3a1. S'acaba el cas d'ús</div>			

Cas d'ús	#11 Canviar idioma (usuari registrat)	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema.		
Disparador	L'usuari vol canviar l'idioma de l'aplicació		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol canviar l'idioma de l'aplicació des de la vista de “Configuració”.</div> <div>2. El sistema canvia l'idioma de l'aplicació per la sessió actual.</div> <div>3. El sistema es comunica amb l'API, que canvia l'idioma del usuari donat a la BD.</div> <div>4. El sistema redirigeix a l'usuari a la vista principal.</div>			
Extensions			
-			

Cas d'ús	#12 Activar mode sense spoilers	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema.		

Disparador	L'usuari vol activar el mode sense spoilers
Escenari principal d'èxit	
<ol style="list-style-type: none"> 1. L'usuari indica al sistema que vol activar el mode sense spoilers. 2. El sistema canvia el mode de l'aplicació per la sessió actual. 3. El sistema es comunica amb l'API, que canvia el mode del usuari donat a la BD. 4. El sistema redirigeix a l'usuari a la vista principal. 	
Extensions	
-	

Cas d'ús	#13 Afegir equip a preferits	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema. L'usuari no ha afegit aquest equip a preferits.		
Disparador	L'usuari vol afegir un equip a preferits.		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol afegir un equip a preferits.</div> <div>2. El sistema es comunica amb l'API, que afegeix l'equip donat als preferits de l'usuari donat.</div> <div>3. Es marca la icona d'equip preferit de l'equip en qüestió.</div>			
Extensions			
-			

Cas d'ús	#14 Eliminar equip de preferits	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema. L'usuari ha afegit aquest equip a preferits.		
Disparador	L'usuari vol eliminar un equip de preferits.		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol eliminar un equip de preferits.</div> <div>2. El sistema es comunica amb l'API, que afegeix l'equip donat als preferits de l'usuari donat.</div> <div>3. Es desmarca la icona d'equip preferit de l'equip en qüestió.</div>			
Extensions			
-			

Cas d'ús	#15 Veure perfil propi	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema.		
Disparador	L'usuari vol veure el seu perfil.		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol veure el seu perfil.</div> <div>2. El sistema mostra a l'usuari les dades de la sessió actual.</div>			
Extensions			

-

Cas d'ús	#16 Editar nom d'usuari	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema.		
Disparador	L'usuari està en el seu perfil i vol editar el seu nom d'usuari.		
Escenari principal d'èxit			
<div>1. L'usuari introdueix un nom.</div> <div>2. L'usuari indica al sistema que vol canviar el seu nom d'usuari per l'indicat.</div> <div>3. El sistema comprova que el nom compleixi els requisits.</div> <div>4. El sistema es comunica amb l'API, que modifica el nom d'usuari de l'usuari donat.</div> <div>5. El sistema modifca la sessió actual amb el nou nom d'usuari.</div> <div>6. El sistema indica a l'usuari que el nom s'ha modificat correctament.</div>			
Extensions			
<div>3a. El nom és el mateix que l'anterior.</div> <div>3a1. El sistema indica a l'usuari que el nom no pot ser el mateix que ja tenia.</div> <div>3a2. S'acaba el cas d'ús.</div> <div>3b. El nom és buit.</div> <div>3b1. El sistema indica a l'usuari que el nom no pot ser buit.</div> <div>3b2. S'acaba el cas d'ús.</div> <div>3c. El nom conté caràcters no alfanumèrics.</div> <div>3c1. El sistema indica a l'usuari que el nom no pot contenir caràcters alfanumèrics.</div> <div>3c2. S'acaba el cas d'ús.</div> <div>3d. El nom té més de 20 caràcters.</div> <div>3d1. El sistema indica a l'usuari que el nom no pot tenir més de 20 caràcters.</div> <div>3d2. S'acaba el cas d'ús.</div>			

Cas d'ús	#17 Enviar comentari	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema.		
Disparador	L'usuari està a la pàgina d'una sèrie i vol enviar un comentari.		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol enviar un comentari.</div> <div>2. El sistema mostra un formulari a l'usuari.</div> <div>3. L'usuari escriu el seu comentari.</div> <div>4. L'usuari prem “Enviar”.</div> <div>5. El sistema comprova que el comentari compleixi els requisits.</div> <div>6. El sistema es comunica amb l'API, que crea el nou comentari a BD.</div> <div>7. El sistema es comunica amb l'API, que retorna la llista dels comentaris de la sèrie donada ordenada per data.</div> <div>8. El sistema mostra la llista dels comentaris.</div>			
Extensions			
<div>4a. L'usuari prem “Cancel·la”.</div> <div>4a1. S'acaba el cas d'ús.</div> <div>5a. El comentari és buit.</div> <div>5a1. El sistema indica a l'usuari que el comentari no pot ser buit.</div>			

Cas d'ús	#18 Votar comentari positivament	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema.		

Disparador	L'usuari està a la pàgina d'una sèrie i vol votar positivament un comentari.
Escenari principal d'èxit	
<ol style="list-style-type: none"> 1. L'usuari indica al sistema que vol votar positivament un comentari. 2. El sistema comprova que l'usuari no havia votat ni positivament ni negativament el comentari. 3. El sistema es comunica amb l'API, que augmenta en un la puntuació del comentari donat. 4. El sistema actualitza la puntuació del comentari en pantalla. 5. El sistema acoloreix el botó de vot positiu del comentari. 	
Extensions	
<ol style="list-style-type: none"> 2a. L'usuari ja havia votat positivament el comentari. <ol style="list-style-type: none"> 2a1. El sistema es comunica amb l'API, que disminueix en un la puntuació del comentari donat. 2a2. El sistema actualitza la puntuació del comentari en pantalla. 2a3. El sistema descoloreix el botó de vot positiu del comentari. 2a4. S'acaba el cas d'ús. 2b. L'usuari ja havia votat negativament el comentari. <ol style="list-style-type: none"> 2a1. El sistema es comunica amb l'API, que augmenta en dos la puntuació del comentari donat. 2a2. El sistema actualitza la puntuació del comentari en pantalla. 2a3. El sistema descoloreix el botó de vot negatiu del comentari. 2a4. El sistema acoloreix el botó de vot positiu del comentari. 2a5. S'acaba el cas d'ús. 	

Cas d'ús	#19 Votar comentari negativament	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema.		
Disparador	L'usuari està a la pàgina d'una sèrie i vol votar negativament un comentari.		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol votar negativament un comentari.</div> <div>2. El sistema comprova que l'usuari no havia votat ni positivament ni negativament el comentari.</div> <div>3. El sistema es comunica amb l'API, que disminueix en un la puntuació del comentari donat.</div> <div>4. El sistema actualitza la puntuació del comentari en pantalla.</div> <div>5. El sistema acoloreix el botó de vot negatiu del comentari.</div>			
Extensions			
<div>2a. L'usuari ja havia votat negativament el comentari.</div> <div>2a1. El sistema es comunica amb l'API, que augmenta en un la puntuació del comentari donat.</div> <div>2a2. El sistema actualitza la puntuació del comentari en pantalla.</div> <div>2a3. El sistema descoloreix el botó de vot negatiu del comentari.</div> <div>2a4. S'acaba el cas d'ús.</div> <div>2b. L'usuari ja havia votat positivament el comentari.</div> <div>2a1. El sistema es comunica amb l'API, que disminueix en dos la puntuació del comentari donat.</div> <div>2a2. El sistema actualitza la puntuació del comentari en pantalla.</div> <div>2a3. El sistema descoloreix el botó de vot positiu del comentari.</div> <div>2a4. El sistema acoloreix el botó de vot negatiu del comentari.</div> <div>2a5. S'acaba el cas d'ús.</div>			

Cas d'ús	#20 Respondre a comentari	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema.		
Disparador	L'usuari vol respondre a un comentari.		
Escenari principal d'èxit			

<ol style="list-style-type: none"> 1. L'usuari indica al sistema que vol respondre a un comentari. 2. El sistema mostra un formulari a l'usuari. 3. L'usuari escriu el seu comentari. 4. L'usuari prem "Enviar". 5. El sistema comprova que el comentari compleixi els requisits. 6. El sistema es comunica amb l'API, que crea el nou comentari a BD. 7. El sistema es comunica amb l'API, que retorna la llista dels comentaris de la sèrie donada ordenada per data. 8. El sistema mostra la llista dels comentaris.
Extensions
<ol style="list-style-type: none"> 4a. L'usuari prem "Cancel·la". <ol style="list-style-type: none"> 4a1. S'acaba el cas d'ús. 5a. El comentari és buit. <ol style="list-style-type: none"> 5a1. El sistema indica a l'usuari que el comentari no pot ser buit.

Cas d'ús	#21 Editar comentari	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema. El comentari a editar és de l'usuari		
Disparador	L'usuari vol editar un comentari seu.		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol editar el seu comentari.</div> <div>2. El sistema mostra un formulari a l'usuari.</div> <div>3. L'usuari escriu el seu comentari.</div> <div>4. L'usuari prem “Enviar”.</div> <div>5. El sistema comprova que el comentari compleixi els requisits.</div> <div>6. El sistema es comunica amb l'API, que actualitza el comentari a BD.</div> <div>7. El sistema modifica el comentari que es mostra en pantalla.</div>			
Extensions			
<div>4a. L'usuari prem “Cancel·la”.</div> <div>4a1. S'acaba el cas d'ús.</div> <div>5a. El comentari és buit.</div> <div>5a1. El sistema indica a l'usuari que el comentari no pot ser buit.</div>			

Cas d'ús	#22 Eliminar comentari propi	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema. El comentari a eliminar és de l'usuari.		
Disparador	L'usuari vol eliminar un comentari seu.		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol eliminar el seu comentari.</div> <div>2. El sistema pregunta a l'usuari si realment vol eliminar el comentari.</div> <div>3. L'usuari respon que sí.</div> <div>4. El sistema es comunica amb l'API, que elimina el comentari donat de la BD.</div> <div>5. El sistema es comunica amb l'API, que retorna els comentaris de la sèrie donada.</div> <div>6. El sistema mostra els comentaris per pantalla.</div>			
Extensions			
<div>4a. L'usuari respon que no.</div> <div>4a1. S'acaba el cas d'ús.</div>			

Cas d'ús	#23 Filtrar sèries per equips preferits	Actor principal	Usuari registrat
Precondicions	L'usuari ha iniciat sessió en el sistema		
Disparador	L'usuari vol filtrar les sèries per els seus equips preferits		
Escenari principal d'èxit			
<div>5. L'usuari indica al sistema que vol veure les sèries dels seus equips preferits.</div> <div>6. El sistema es comunica amb l'API, que retorna una llista amb les properes sèries i una llista amb les sèries acabades demanades.</div> <div>7. El sistema guarda les sèries rebudes.</div> <div>8. El sistema redirigeix a l'usuari a la vista de "Llista de sèries", on es mostren les properes sèries en una pestanya i les acabades en una altra.</div>			
Extensions			
<div>2a. Alguna de les llistes retornades és buida.</div> <div>2a1. El sistema indica a l'usuari que no s'ha trobat cap sèrie.</div>			

Cas d'ús	#24 Eliminar comentari	Actor principal	Moderador
Precondicions	El moderador ha iniciat sessió en el sistema		
Disparador	El moderador vol eliminar un comentari.		
Escenari principal d'èxit			
<div>1. El moderador indica al sistema que vol eliminar el seu comentari.</div> <div>2. El sistema pregunta a l'usuari si realment vol eliminar el comentari.</div> <div>3. El moderador respon que sí.</div> <div>4. El sistema es comunica amb l'API, que elimina el comentari donat de la BD.</div> <div>5. El sistema es comunica amb l'API, que retorna els comentaris de la sèrie donada.</div> <div>6. El sistema mostra els comentaris per pantalla.</div>			
Extensions			
<div>4a. El moderador respon que no.</div> <div>4a1. S'acaba el cas d'ús.</div>			

Cas d'ús	#25 Actualitzar BD	Actor principal	Administrador de BD
Precondicions	-		
Disparador	L'administrador vol actualitzar la BD amb l'última informació		
Escenari principal d'èxit			
<div>1. L'administrador indica al sistema que vol actualitzar la BD.</div> <div>2. El sistema cerca totes les competicions actives i properes a l'API externa.</div> <div>3. Per a cada competició:<div>a. Per a cada equip de la competició:<div>i. S'actualitza la informació de l'equip si és necessari.</div><div>ii. S'actualitzen els jugadors de l'equip si és necessari.</div></div><div>b. S'afegeixen les properes sèries de la competició a la BD.</div></div>			
Extensions			
<div>3a. La competició no existeix a BD.</div> <div>3a1. S'afegeix la competició a la BD.</div> <div>3a_a. L'equip no existeix a BD.</div>			

3a_a1. S'afegeix l'equip a la BD.
3a_b. L'equip ja no forma part de la competició.
3a_b1. S'elimina l'equip de la competició a BD.
3a_c. L'equip no formava part de la competició a BD.
3a_c1. S'afegeix l'equip a la competició a BD.
3ai_a. El jugador no existeix a BD.
3ai_a1. S'afegeix el jugador a la BD.

Cas d'ús	#26 Afegir resultats a BD	Actor principal	Administrador de BD
Precondicions	Hi ha com a mínim una competició activa.		
Disparador	L'administrador vol posar els últims resultats a BD		
Escenari principal d'èxit			
<div>1. L'administrador indica al sistema que vol posar els últims resultats a BD.</div> <div>2. El sistema troba totes les competicions actives a BD.</div> <div>3. Per a cada competició:<div>a. El sistema troba les sèries acabades de la competició que no estàn a BD.</div><div>b. Per a cada sèrie:<div>i. El sistema troba la sèrie corresponent a BD.</div><div>ii. El sistema recolleix les dades de la sèrie i actualitza la sèrie tobadada a BD.</div><div>iii. El sistema recolleix les dades de cada partida de la sèrie i la crea a BD.</div></div></div>			
Extensions			
<div>3bi_a. La sèrie no existeix a BD.</div> <div>3bi_a1. El sistema recolleix les dades de la sèrie i crea la sèrie a BD.</div> <div>3bi_a2. Es retorna al pas 3bii del cas d'ús.</div> <div>3bii_a. Un dels equips no existeix a BD.</div> <div>3aii_a1. S'afegeix l'equip a la BD.</div> <div>3bii_b. Un dels jugadors no existeix a BD.</div> <div>3aii_b1. S'afegeix el jugador a la BD.</div>			

7.4. Model conceptual de les dades

El model conceptual de les dades representa l'estructura de les dades requerida per dur a terme els processos de negoci sense tenir en compte la implementació interna o les estructures d'emmagatzematge de dades [17].

En aquest projecte, les classes més importants són “Series” i “User”, seguides per “Team”, “Tournament” i “Player”.

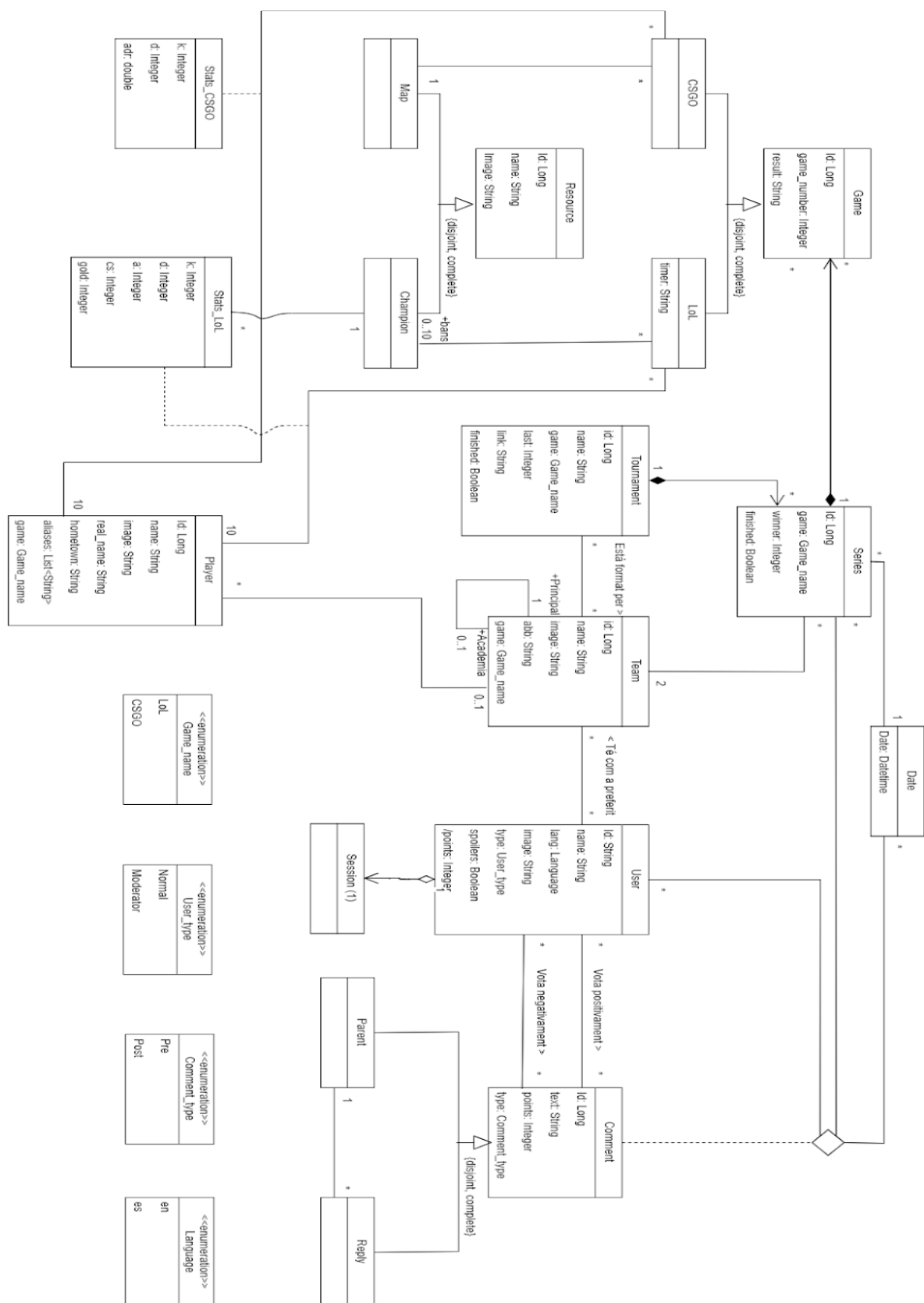


Figura 10. Model conceptual de les dades
Font: Elaboració pròpia

7.4.1. Restriccions d'integritat

Tot seguit es llisten totes aquelles restriccions i condicions que no es poden representar a l'esquema.

- Claus primàries: (Game, id), (Resource, id), (Series, id), (Date, date), (Tournament, id), (Team, id), (Player, id), (User, id), (Comment, id).
- Finished de Serie no pot ser true si Date > data actual.
- Winner de Serie es null sempre que Finished sigui fals.
- Finished de Tournament no pot ser true si no han acabat totes les seves series.
- Un equip no pot tenir-se a si mateix per acadèmia.
- Un usuari no pot votar positivament i negativament el mateix comentari a la vegada.
- Un comentari no pot ser creat en una data superior a l'actual.
- Una partida (Game) serà de LoL si la seva sèrie té "LoL" com a game, i de CSGO si té "CSGO" com a game.

7.42. Descripció de les classes

Per tenir una millor comprensió del model, tot seguit es detallen les classes del model amb els seus atributs i es justifiquen algunes de les decisions preses.

Series

Representa enfrontament competitiu entre dos equips del sistema, és a dir, un conjunt de partides entre els dos mateixos equips. La data indica el temps de començament de la sèrie.

- id. Identificador.
- game. Nom del joc de la sèrie.
- winner. Indica quin dels dos equips ha guanyat la sèrie (1 per el local i 2 per el visitant).
- finished. Indica si la sèrie ha finalitzat

Data

Representa una data.

- date. Datetime en la zona horària GMT+0

Game (Partida)

Representa una partida que forma part d'una sèrie. S'especialitza en LoL i CSGO per la diferència dels atributs entre les subclasses (mentres que en series no cal perquè ambdós tenen els mateixos atributs).

- id. Identificador.
- game_number. Número de la partida dins de la sèrie. Per exemple, si la sèrie té 5 partides, hi haurà 5 partides amb game_numbers entre el 1 i el 5.
- result. Resultat de la partida. Per exemple, "1-0".
- timer. Duració de la partida. Per exemple, "24:51".

Resource

Representa un element de les partides, Champions (a partir d'ara campions, són personatges) per partides de LoL i Maps (a partir d'ara mapes) per partides de CSGO. Tot i que els dos tenen els mateixos atributs s'ha decidit especialitzar en subclasses per claredat. Els bans d'una partida de LoL representen els personatges bloquejats de la partida.

- id. Identificador.
- name. Nom del personatge o mapa.
- image. Imatge del personatge o mapa.

Player

Representa un jugador professional. S'ha decidit que necessitava un identificador perquè és possible que hi hagi dos jugador amb el mateix nom.

- id. Identificador.
- name. Nickname del jugador.
- image. Imatge del jugador.
- real_name. Nom real del jugador.
- hometown. Lloc de procedència del jugador.
- aliases. Llista de nicknames alternatius amb els que ha jugat.
- game. Joc en el que el jugador juga professionalment actualment.

Stats_LoL

Representa les estadístiques d'un jugador en una partida de LoL amb un personatge determinat.

- k. Assassinats del jugador.
- d. Morts del jugador.
- a. Assistències del jugador.
- cs. "Creeps" del jugador.
- gold. Or del jugador.

Stats_CSGO

Representa les estadístiques d'un jugador en una partida de CSGO. En aquest cas no té una connexió amb Map (al contrari de Stats_LoL, que si en té amb campió) perquè una partida de CSGO només té un mapa, mentre que cada jugador d'una partida de LoL juga amb un campió diferent.

- k. Assassinats del jugador.
- d. Morts del jugador.
- adr. "Average Damage per Round" del jugador, és a dir, el mal que fa de mitjana per ronda el jugador.

Tournament

Representa una competició del sistema, que està formada per diverses series i equips.

- id. Identificador.
- name. Nom de la competició.
- game. Nom del joc al qual pertany la competició.
- last. Últim índex de sèrie processada per ordre cronològic (útil per no afegir la mateixa sèrie a BD dues vegades en cas de que encara no s'hagin jugat totes.).

- link. Link amb la llista de les sèries.
- finished. Indica si la competició s'ha acabat.

Team

Representa un equip en el sistema. L'associació recursiva representa l'acadèmia de l'equip (si en té).

- id. Identificador.
- name. Nom de l'equip.
- image. Imatge de l'equip.
- abb. Abreviatura de l'equip.
- game. Joc en el que competeix l'equip.

User

Representa un usuari en el sistema. Pot tenir com a preferits diversos equips.

- id. Identificador de Google
- name. Nom de l'usuari.
- lang. Llengua del sistema de l'usuari.
- image. Imatge de perfil de l'usuari.
- type. Indica si l'usuari té permisos de moderador.
- spoilers. Indica si l'usuari vol amagar els spoilers.
- /points. Atribut calculat. Suma dels punts dels comentaris de l'usuari.

Session

Classe singleton que representa l'usuari que ha iniciat sessió actualment en el sistema.

Comment

Representa un comentari en el sistema. Un comentari està format per un usuari, una sèrie i una data de creació. Un comentari pot ser Pare (és a dir, respon a la sèrie directament), o rèplica (és a dir, respon a un comentari pare). A més un comentari pot ser votat positivament i negativament per diversos usuaris.

- id. Identificador del comentari.
- text. Missatge del comentari.
- points. Puntuació del comentari.
- type. Indica si el comentari s'ha fet a la prèvia de la sèrie o en el seu post-partit.

7.5. Requisits no funcionals

En aquest apartat es llisten els diferents requisits no funcionals que ha de complir l'aplicació. Per facilitar en l'organització d'aquests requisits, s'ha basat la llista en la classificació de requisits no funcionals de Volere [18].

Per a cada requisit s'especifica la classificació del requisit segons Volere, una descripció del requisit i el seu criteri d'acceptació.

11a. Ease of use Requirement. El sistema ha de ser fàcil d'utilitzar per a qualsevol persona amb un coneixement mínim d'aplicacions mòbil i de competicions d'*e-sports*.

Criteri d'acceptació: El 80% dels usuaris enquestats han de ser capaços de completar les tasques bàsiques de l'aplicació sense informació prèvia.

11b. Personalization and Internationalization Requirement. El sistema ha d'estar disponible en diverses llengües.

Criteri d'acceptació: L'aplicació estarà disponible en castellà i anglès.

11b. Personalization and Internationalization Requirement. El sistema ha de recordar la configuració bàsica de l'usuari.

Criteri d'acceptació: L'aplicació guarda la configuració bàsica de l'usuari (idioma i mode spoilers) en el 100% dels casos.

11b. Personalization and Internationalization Requirement. Les dates i hores han d'estar en la zona horària de l'usuari

Criteri d'acceptació: L'aplicació mostra les hores en la zona horària de l'usuari. Si l'usuari canvia la zona horària del dispositiu, l'aplicació reflecteix els canvis.

11d. Understandability and Politeness Requirement. Els termes utilitzats per el sistema han de ser comprensibles per la comunitat d'usuaris a la que orientat el sistema.

Criteri d'acceptació: El 80% dels usuaris que formen part de la comunitat d'usuaris a la que va destinada l'aplicació (és a dir, els usuaris que han jugat als jocs que ofereix), entenen els termes específics que s'utilitzen.

11e. Accessibility Requirement. El sistema ha de ser usable persones amb diferents discapacitats visuals, com pot ser el daltonisme.

Criteri d'acceptació: Diverses persones amb discapacitats visuals poden utilitzar el sistema sense dificultats.

12a. Speed and Latency Requirement. El sistema no trigarà més de 5 segons en mostrar cap pantalla amb una connexió superior als 10Mb/s.

Criteri d'acceptació: Es farà un test del que triga en carregar les diverses pantalles amb diverses connexions a Internet.

12d. Robustness or Fault-Tolerance Requirement. En cas d'error, l'aplicació no haurà de penjar-se.

Criteri d'acceptació: L'aplicació no es penja en casos extrems.

13b. Requirement for Interfacing with Adjacent Systems. L'aplicació ha de funcionar en la gran majoria de dispositius Android.

Criteri d'acceptació: L'aplicació ha de funcionar com a mínim en el 80% dels dispositius Android.

8. Arquitectura del sistema

En aquest capítol es presenta i es justifica l'arquitectura i el disseny de les diverses parts del sistema: l'aplicació mòbil, les bases de dades, les APIs i les connexions amb els diferents serveis externs necessaris pel correcte funcionament del sistema.

La secció 8.1 dóna una visió global de l'arquitectura del sistema. En la secció 8.2 es descriu el disseny del back-end, és a dir, el disseny de la base de dades pròpia i de l'API REST que hi dóna accés. Després, en la secció 8.3, es descriu el disseny del front-end, és a dir, el disseny de la interfície, de la capa de presentació i de les classes del disseny. Per últim, en la secció 8.4 es mostren els patrons utilitzats i en la 8.5 es mostren els diagrames de registre més representatius del sistema.

8.1. Visió global

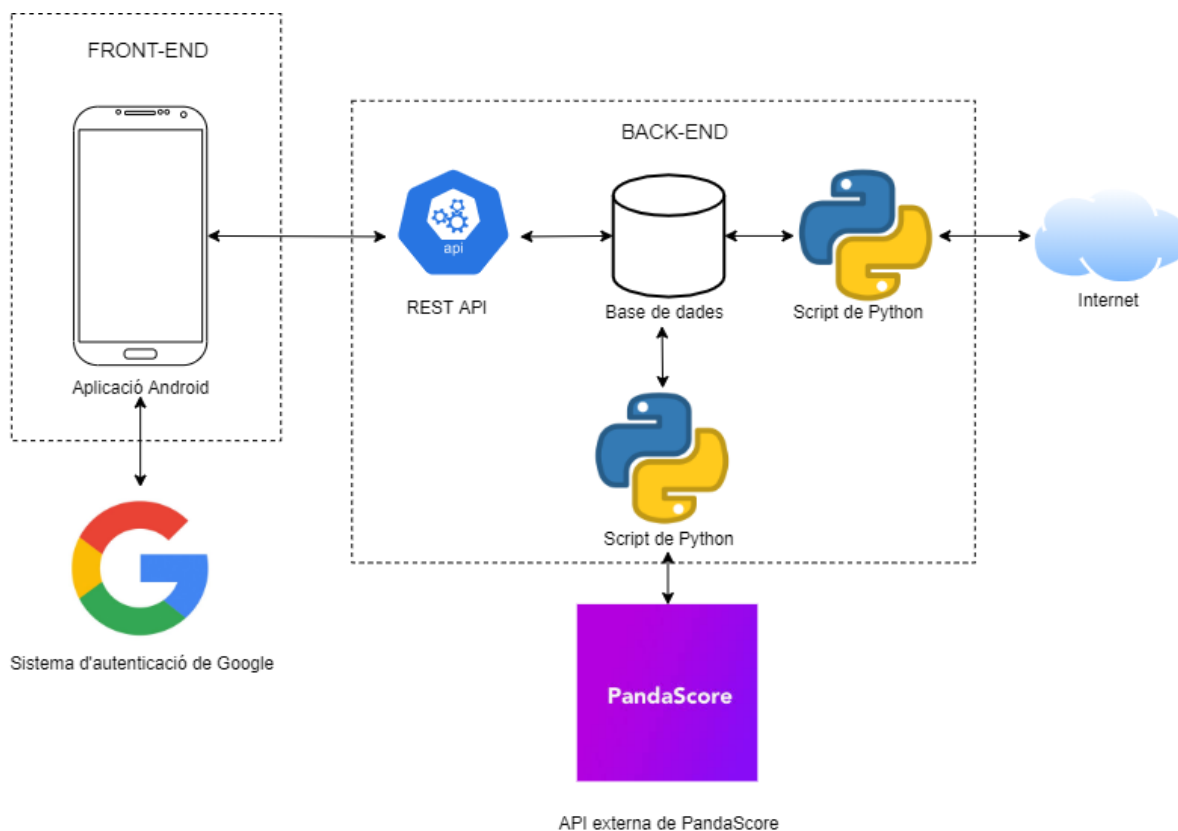


Figura 11. Visió global de l'arquitectura del sistema
Font: Elaboració pròpia

Tal com es mostra a la figura 11, el sistema té diversos components importants que s'han dissenyat i desenvolupat: l'aplicació Android, la base de dades, la REST API i dos scripts de Python.

Per una banda, l'aplicació Android, que és el component principal del sistema i amb el que interactuen els usuaris, es comunica amb el sistema d'autenticació de Google per tal que els usuaris es puguin registrar al sistema i iniciar sessió. He decidit fer només l'aplicació per a Android per diversos motius. El més important és que ja tenia experiència prèvia desenvolupant aplicacions per a Android, mentre que mai he fet res per a iOS o altres sistemes operatius de dispositius mòbils. Al principi, es volia fer amb Ionic, una eina que permet desenvolupar aplicacions per a tots els dispositius mòbils a la vegada, però es va desestimar perquè es va considerar que el temps d'aprenentatge seria massa elevat.

D'altra banda, les dades del sistema es guarden en una base de dades no relacional MongoDB [19].

Les bases de dades més típiques són les que segueixen un model relacional, és a dir, aquelles que tenen un esquema marcat i fix. Aquesta és una característica que no és compatible amb aquest sistema, ja que es vol tenir la possibilitat de guardar dades diferents en un futur d'una manera fàcil i la possibilitat d'afegir noves dades sense la necessitat de fer un canvi en l'esquema. Per aquest motiu s'ha triat una base de dades no relacional basada en documents. Dins de les bases de dades documentals, hi ha diversos productes, però s'ha triat MongoDB per la meua experiència prèvia i per l'existència d'Atlas MongoDB (un servei que permet allotjar bases de dades MongoDB al núvol) [20], on és allotjada. S'ha triat aquest servei per tal de facilitar l'accés a la base de dades des de qualsevol lloc i perquè permet la possibilitat de fer consultes ràpides mitjançant la seva interfície web.

Aquesta base de dades es comunica amb l'API externa de PandaScore [21] mitjançant un script de Python, que s'encarrega d'aconseguir la informació bàsica del sistema disponible a PandaScore, com poden ser les competicions actuals, els equips que les formen o els canvis de jugadors entre equips.

Es van trobar un total de 3 APIs externes que proporcionaven dades i informació sobre competicions d'*e-sports*. Una d'elles ni tan sols tenia documentació i no van contestar els mails enviats demanant informació sobre les funcionalitats i el preu, pel que va quedar descartada. Entre les altres dos, es va escollir finalment PandaScore per la qualitat de la seva documentació i perquè tenia la base de dades més completa i actualitzada. A més, PandaScore disposa de dades i estadístiques de més *e-sports*, per si en un futur es decideix afegir-los en el sistema. L'únic problema és que la versió gratuïta no ofereix tota la informació necessària per a garantir que es compleixen els objectius d'aquest sistema, i el preu de la versió bona és inassumible per part meua en aquests moments.

Per aquest motiu, un altre script de Python s'encarrega d'omplir la base de dades amb les dades que no estan disponibles a PandaScore, com els resultats i les estadístiques de les partides. Per a fer-ho, agafa la informació de pàgines externes mitjançant *HTML Scraping*, una tècnica d'extracció de dades que s'explica en profunditat a l'apartat 9.3.2 d'aquesta memòria.

Aquests dos scripts s'han fet amb el llenguatge de programació Python sobretot per la meua experiència prèvia, ja que és el llenguatge que personalment més domino, però també per

la bona qualitat de la llibreria *pymongo*, que serveix per afegir i modificar dades d'una base de dades MongoDB amb Python. Inicialment, es planejava que aquests scripts s'executessin automàticament cada 8 hores, però de moment s'executen manualment una vegada al dia. Això és degut a les dificultats trobades a l'hora d'aconseguir executar automàticament l'script mitjançant Heroku.

Per últim, s'ha desenvolupat una API per tal de que l'aplicació pugui comunicar-se amb la base de dades. La millor opció per a aquest projecte és la de desenvolupar una REST API. Hi ha altres tipus d'APIs, com pot ser el cas d'una API que utilitzi SOAP, que són més segures que les de tipus REST, però també són més complexes, fent que el procés d'aprenentatge sigui més alt que no amb una REST API. Com que la meua experiència fent APIs es limita a les de tipus REST, es va descartar fer una de tipus SOAP.

Les REST API estan basades en dades, pel que són compatibles amb el format JSON, fet que facilita molt la feina a l'hora de retreure les dades demanades. A més, les REST API estan basades en URLs, és a dir, retreuen dades quan dones l'URI pertinent, pel que segueix el protocol HTTP i possibilita fer crides a l'API des d'un dispositiu mòbil fàcilment. En resum, s'ha triat fer servir una REST API per la seva senzillesa en comparació a altres tipus d'APIs i perquè són compatibles amb el format de dades JSON.

Aquesta API està desplegada en un servidor de Heroku, per tal de garantir la màxima disponibilitat possible. De la seva pàgina web, Heroku és "una plataforma com a servei (PaaS) que permet a desenvolupadors construir, executar i operar aplicacions totalment al núvol" [22]. El punt fort d'aquí és la part que s'opera al núvol, fet que permet accedir a l'aplicació des de qualsevol lloc en qualsevol moment sense preocupar-se de mantenir un servidor a casa. A més, el pla gratuït permet utilitzar l'aplicació un total de 450 hores per més, que considero prou per a aquest projecte, tot i que la idea en el futur és pagar per tenir hores il·limitades o trobar una alternativa millor.

Tot i que la meua experiència prèvia fent REST APIs era amb Swagger, un servei que permet construir i desplegar RESTful APIs, s'ha fet amb Python per ser consistent amb els dos scripts, de manera que tot el back-end s'ha desenvolupat amb Python.

8.2. Disseny del back-end

Tot i que el més comú en projectes d'aquest estil és dissenyar primer el front-end, s'ha decidit que, a causa de la importància del tractament de les dades en aquest projecte, és més raonable començar pel disseny de la base de dades i adaptar les classes de l'aplicació a aquest disseny.

8.2.1. Normalització del diagrama de classes

Primer, tenint com a base el model conceptual de les dades mostrat a la figura 10, cal dissenyar un diagrama de les classes de la base de dades.

Tal com es mostra a la figura 12, aquest diagrama és bastant similar al del model conceptual de les dades, però més simplificat, eliminant certes classes com la data o juntant subclasses com Champion i Map en Resource, o Parent i Reply en Comment. Per altra banda, la classe Game ha deixat d'existir, per la diferència tan gran d'atributs que tenen Game_lol i Game_csgo (en el proper apartat s'explica més detalladament els atributs de les classes).

Aquest disseny s'ha fet tenint en compte les dades que es poden extreure de l'API externa de PandaScore i les dades que seran necessàries per a l'aplicació mòbil. Per exemple, PandaScore ens permet saber els jugadors d'un equip, pel que el més fàcil és tenir una relació normal entre equip i jugador, però no ens permet saber els jugadors d'una partida (o sèrie) en concret, pel que el millor és posar una associació entre jugador i les partides per tal que sèrie també pugui navegar fins als jugadors d'una partida en concret.

8.2.2. Disseny lògic de la base de dades

Com s'ha comentat abans, la base de dades és una base de dades no relacional feta amb MongoDB. Una base de dades no relacional és "una base de dades que no utilitza un esquema de taules compostes per files i columnes. En comptes, les bases de dades no relacionals tenen un model d'emmagatzematge optimitzat pels requisits específics de les dades emmagatzemades, com poden ser parells de key/value, documents JSON o grafs", segons la documentació de Microsoft [23].

En el cas de MongoDB s'utilitzen documents JSON, pel que el format del disseny lògic mostrat a continuació serà aquest. Per a cada col·lecció de la base de dades, tot seguit s'inclou un exemple concret de la base de dades amb algunes dades, s'especifiquen els atributs d'un dels seus documents i s'expliquen aquells que no estiguin especificats en el model conceptual de les dades:

Series

```
_id: 545232
game: "LoL"
date: "2019-08-10"
tournament: Object
  id: 2380
  name: "CBLol - Winter 2019 - Regular season"
  format: 1
team1: Object
  id: 2884
  name: "UpperCut Esports"
  abb: "UP"
  wins: 1
  image: "https://cdn.pandascore.co/images/team/image/2884/idm-gaming-fteacgbn.p..."
team2: Object
  id: 123146
  name: "Redemption eSports"
  abb: "RDP"
  wins: 0
  image: "https://cdn.pandascore.co/images/team/image/123146/redemption-esports-..."
finished: 1
time: "00:00"
```

{

 _id: *Long*,

 game: *String*,

 date: *String*,

 tournament: //Objecte que referència la competició de la sèrie

 {

 id: *Long*,

 name: *String*

 },

 team1: //Objecte que referència l'equip local de la sèrie

 {

 id: *Long*,

```

        name: String,
        abb: String,
        image: String
    },
    team2: //Objecte que referència l'equip visitant de la sèrie
    {
        id: Long,
        name: String,
        abb: String,
        image: String
    },
    finished: Boolean,
    time: String
}

```

Tournament

```

    _id: 2528
    name: "World Championship - 2019 - Group C"
    game: "LoL"
    teams: Array
    team0: Object
    team1: Object
    team2: Object
    team3: Object
    last: 50
    link: "https://lol.gamepedia.com/2019_Season_World_Championship/Main_Event/Ma..."
    finished: true
    series: Array
    series0: 548755
}

_id: Long,
game: String,
teams: //Array d'objectes que referencien els equips de la competició.
[
    {
        id: Long,
        name: String
    }
],
last: Integer,
link: String,
finished: Boolean
}

```

Team

```
_id: 394
name: "Fnatic"
game: "LoL"
image: "https://cdn.pandascore.co/images/team/image/394/Fnaticlogo-square.png"
▼ players: Array
  ▼ 0: Object
    id: 74
    name: "Hylissang"
    country: "Bulgaria"
  > 1: Object
  > 2: Object
  > 3: Object
  > 4: Object
▼ tournaments: Array
  ▼ 0: Object
    id: 2438
    name: "LEC - Summer 2019 - Playoffs"
    place: null
  > 1: Object
  > 2: Object
  > 3: Object
  > 4: Object
  > stats: Array
  > series: Array
    abb: "FNC"
  > academy: Object
```

{

```
_id: Long,
name: String,
game: String,
image: String,
players: //Array d'objectes que referencien els jugadors de l'equip.
[
  {
    id: Long,
    name: String,
    hometown: String
  }
],
tournaments: //Array d'objectes que referencien les competicions de l'equip.
[
  {
    id: Long
    name: String
  }
],
stats: //Array d'objectes que referencien les estadístiques de l'equip per a cada
```

Resource

```
[
  { //Exemple d'un equip de LoL
```



```

    champion: //Objecte que referència un campió
    {
        id: Long,
        name: String,
        image: String,
    },
    games: Integer, //Nombre de partides jugades
    wins: Integer, //Nombre de victòries
    loses: Integer, //Nombre de derrotes
    k: Integer,
    d: Integer,
    a: Integer,
    cs: Integer,
    gold: Integer
},
{ //Exemple d'un equip de CSGO
    map: //Objecte que referència un resource Map
    {
        id: Long,
        name: String
    },
    games: Integer, //Nombre de partides jugades
    wins: Integer, //Nombre de victòries
    loses: Integer, //Nombre de derrotes
    round_wins: Integer //Nombre de rondes guanyades
    round_loses: Integer //Nombre de rondes perdudes
    k: Integer,
    d: Integer,
    adr: Double
}
],
series: List<Long>, //Array amb els identificadors de les sèries
abb: String,
academy: //Objecte que referència l'acadèmia de l'equip
{
    team: //Objecte que referència l'equip acadèmia
    {
        id: Long,
        name: String,
        abb: String,
    },
    players: //Array d'objectes que referencien els jugadors de l'equip.
    {
        id: Long,
        name: String,
        hometown: String
    }
}

```

```

    }
  }
}

```

Player

```

_id: 8273
name: "Spica"
real_name: "Ming Lu"
game: "LoL"
image: "https://cdn.pandascore.co/images/player/image/8273/spica-9jc81kro.png"
country: null
slug: "spica"
team: null
series: Array
  0: 547343
  1: 10000000259
  2: 10000000263
stats: Array
  0: Object
    c_id: 1097
    c_name: "Sejuani"
    c_image: "https://cdn.pandascore.co/images/lol/champion/image/334bbb28d99b00df03..."
    games: 7
    wins: 3
    loses: 4
    k: 8
    d: 22
    a: 41
    cs: 1027
    gold: 67
  1: Object
aliases: Array
{
  _id: Long,
  name: String,
  real_name: String,
  game: String,
  image: String,
  hometown: String,
  team: //Objecte que referència a l'equip del jugador
  {
    id: Long,
    name: String,
    abb: String
  }
  stats: //Array d'objectes que referencien les estadístiques del jugador per a cada
Resource
[
  { //Exemple d'un equip de LoL
    champion: //Objecte que referència un campió
    {
      id: Long,

```

```

        name: String,
        image: String,
    },
    games: Integer, //Nombre de partides jugades
    wins: Integer, //Nombre de victòries
    loses: Integer, //Nombre de derrotes
    k: Integer,
    d: Integer,
    a: Integer,
    cs: Integer,
    gold: Integer
},
{ //Exemple d'un equip de CSGO
    map: //Objecte que referència un resource Map
    {
        id: Long,
        name: String
    },
    games: Integer, //Nombre de partides jugades
    wins: Integer, //Nombre de victòries
    loses: Integer, //Nombre de derrotes
    round_wins: Integer //Nombre de rondes guanyades
    round_loses: Integer //Nombre de rondes perdudes
    k: Integer,
    d: Integer,
    adr: Double
}
],
series: List<Long> //Array que referència els identificadors de les sèries del jugador
aliases: List<String>
}

```

Resource

```
_id: 1000
name: "Aatrox"
slug: "aatrox"
image: "https://cdn.pandascore.co/images/lol/champion/image/df4f3e50d7a305a78b..."
```

```
{
  _id: Long,
  name: String,
  image: String
}
```

Game_lol

```
_id: 601
series_id: 547312
game_number: 1
timer: "31:23"
team1: 325
team2: 1635
team1_stats: Object
  name: "INF"
  winner: 1
  kills: 9
  gold: 60
  turrets: 10
  > bans: Array
team2_stats: Object
  name: "R7"
  winner: 0
  kills: 2
  gold: 49
  turrets: 2
  > bans: Array
players1: Array
  > 0: Object
    id: 3462
    name: "Straight"
    c_id: 1000
    c_name: "Aatrox"
    k: 2
    d: 0
    a: 4
    cs: 265
    gold: 13
  > 1: Object
  > 2: Object
  > 3: Object
  > 4: Object
players2: Array
  > 0: Object
  > 1: Object
  > 2: Object
  > 3: Object
  > 4: Object
```

```

{
  _id: Long,
  series_id: Long, //Referència l'identificador de la sèrie a la que pertany
  game_number: Integer,
  timer: String,
  team1: Long //Referència l'identificador de l'equip local
  team2: Long //Referència l'identificador de l'equip visitant
  team1_stats: //Objecte amb les estadístiques de l'equip local
  {
    name: String,
    winner: Boolean //Indica si aquest equip ha guanyat
    k: Integer,
    gold: Integer,
    bans: List<String> //Llista amb els noms dels campions bloquejats per l'equip
  }
  team2_stats: //Objecte amb les estadístiques de l'equip visitant
  {
    name: String,
    winner: Boolean //Indica si aquest equip ha guanyat
    k: Integer,
    gold: Integer,
    bans: List<String> //Llista amb els noms dels campions bloquejats per l'equip
  }
  players1: //Array amb les estadístiques dels jugadors de l'equip local
  [
    {
      id: Long,
      name: String,
      c_id: Long //Identificador del campió,
      c_name: String //Nom del campió
      k: Integer,
      d: Integer,
      a: Integer,
      cs: Integer,
      gold: Integer
    }
  ],
  players2: //Array amb les estadístiques dels jugadors de l'equip visitant
  [
    {
      id: Long,
      name: String,
      c_id: Long //Identificador del campió,
      c_name: String //Nom del campió
      k: Integer,
      d: Integer,

```

```

    a: Integer,
    cs: Integer,
    gold: Integer
  }
],
}

```

Game_csgo

```

_id: 669
series_id: 548086
game_number: 1
map: Object
  m_id: 2002
  m_name: "Dust2"
team1: 3455
team2: 124524
team1_stats: Object
  name: "Vitality"
  rounds: 8
  winner: 0
team2_stats: Object
  name: "Syman"
  rounds: 16
  winner: 1
players1: Array
  0: Object
    id: 18452
    name: "Zyw0o"
    m_id: 2002
    m_name: "Dust2"
    k: 20
    d: 19
    adr: 91.3
  1: Object
  2: Object
  3: Object
  4: Object
players2: Array
vod: null

```

```

{
  _id: Long,
  series_id: Long, //Referència l'identificador de la sèrie a la que pertany
  game_number: Integer,
  map: //Objecte que referència el mapa de la partida
  {
    id: Long,
    name: String
  }
  team1: Long //Referència l'identificador de l'equip local
  team2: Long //Referència l'identificador de l'equip visitant
  team1_stats: //Objecte amb les estadístiques de l'equip local
  {
    name: String,

```

```

        winner: Boolean //Indica si aquest equip ha guanyat
        rounds: Integer //Indica el nombre de rodnes que ha guanyat l'equip
    }
    team2_stats: //Objecte amb les estadístiques de l'equip visitant
    {
        name: String,
        winner: Boolean //Indica si aquest equip ha guanyat
        rounds: Integer //Indica el nombre de rodnes que ha guanyat l'equip
    }
    players1: //Array amb les estadístiques dels jugadors de l'equip local
    [
        {
            id: Long,
            name: String,
            k: Integer,
            d: Integer,
            adr: Integer,
        }
    ],
    players2: //Array amb les estadístiques dels jugadors de l'equip visitant
    [
        {
            id: Long,
            name: String,
            c_id: Long //Identificador del campió,
            c_name: String //Nom del campió
            k: Integer,
            d: Integer,
            a: Integer,
            cs: Integer,
            gold: Integer
        }
    ],
}

```

User

```
_id: "106001019510120959213"
> comments: Array
  lang: "en"
  name: "Mabaro3009"
  teams: Array
    0: 3213
    1: 88
    2: 114
    3: 1568
  type: 1
  image: "https://lh3.googleusercontent.com/a-/AAuE7mCBiz-wBBbus0QD1Yjd5a2MHxjoI..."
  spoilers: true
```

```
{
  _id: String,
  name: String,
  type: Integer,
  lang: String,
  image: String,
  spoilers: Boolean,
  teams: List<Long> //Llista amb els identificadors dels equips preferits de l'usuari
}
```

Comment

```
_id: "1"
parent_id: null
serie_id: 549414
user_id: "11111111111111111111111111111111"
user_name: "Test User"
text: "test comment"
points: -3
date: "2019-11-27"
time: "10:21:00"
type: "pre"
  upvoted_by: Array
    0: "106001019510120959213"
  downvoted_by: Array
```

```
{
  _id: String,
  parent_id: String, //Referència l'identificador del comentari pare. Es null si el
comentari no és una rèplica
  serie_id: Long, //Referència la serie on s'ha publicat el comentari
  user_id: String, //Referència l'usuari que ha publicat el comentari
  user_name: String,
  text: String,
  points: Integer,
  date: String,
  time: String,
  type: String,
```



```

    upvoted_by: List<String> //Llista amb els identificadors dels usuaris que han votat
    positivament aquest comentari
    downvoted_by: List<String> //Llista amb els identificadors dels usuaris que han votat
    negativament aquest comentari
}

```

En alguns casos, s'ha preferit posar informació extra a part de l'identificador per referenciar a altres classes perquè la reducció del nombre de consultes donada per aquest canvi compensa l'augment de la mida del document. Per exemple, en la col·lecció Player poso directament el nom i l'abreviació de l'equip, ja que és informació que serà necessària a l'hora de mostrar l'equip del jugador. D'aquesta manera, només cal fer una consulta, en comptes de dues (primer consultar l'identificador de l'equip del jugador i després consultar el nom d'aquest identificador).

A més, s'ha aprofitat la propietat de MongoDB que permet que els documents d'una mateixa col·lecció no siguin iguals en diverses col·leccions, com pot ser el cas de Player i Team, on "stats" canvia segons si són de LoL o de CSGO.

8.2.3. Disseny de l'API

Com s'ha vist en la visió global del sistema, és necessari desenvolupar una API per tal de comunicar l'aplicació amb la base de dades.

A continuació es mostren les rutes necessàries de l'API d'acord amb els casos d'ús especificats del sistema. Per a cada ruta s'especifica la ruta, el mètode, els atributs d'entrada, una descripció i els possibles valors de retorn.

Ruta	/all/series/upcoming
Mètode	GET
Paràmetres d'entrada	-
Descripció	Retorna les 20 properes series
Retorn	- upcoming_series: List<Serie>

Ruta	/all/series/finished
Mètode	GET
Paràmetres d'entrada	-
Descripció	Retorna les 20 últimes sèries finalitzades
Retorn	- finished_series: List<Serie>

Ruta	/tournaments/<t_id>/series/upcoming
Mètode	GET
Paràmetres d'entrada	- t_id: String - @Path
Descripció	Retorna les 20 properes sèries del tournament especificat
Retorn	- upcoming_series: List<Serie>

Ruta	/tournaments/<t_id>/series/finished
Mètode	GET
Paràmetres d'entrada	- t_id: String - @Path
Descripció	Retorna les 20 últimes sèries finalitzades del tournament especificat
Retorn	- finished_series: List<Serie>

Ruta	/lol/series/upcoming
Mètode	GET
Paràmetres d'entrada	-
Descripció	Retorna les 20 properes sèries de LoL
Retorn	- upcoming_series: List<Serie>

Ruta	/lol/series/finished
Mètode	GET
Paràmetres d'entrada	-
Descripció	Retorna les 20 últimes sèries finalitzades de LoL
Retorn	- finished_series: List<Serie>

Ruta	/csgo/series/upcoming
Mètode	GET
Paràmetres d'entrada	-
Descripció	Retorna les 20 properes sèries de CSGO
Retorn	- upcoming_series: List<Serie>

Ruta	/csgo/series/finished
-------------	-----------------------

Mètode	GET
Paràmetres d'entrada	-
Descripció	Retorna les 20 últimes sèries finalitzades de CSGO
Retorn	- finished_series: List<Serie>

Ruta	/players/<player_id>/series/upcoming
Mètode	GET
Paràmetres d'entrada	- player_id: String - @Path
Descripció	Retorna les 20 properes sèries del jugador especificat
Retorn	- upcoming_series: List<Serie>

Ruta	/players/<player_id>/series/finished
Mètode	GET
Paràmetres d'entrada	- player_id: String - @Path
Descripció	Retorna les 20 últimes sèries finalitzades del jugador especificat
Retorn	- finished_series: List<Serie>

Ruta	/teams/<team_id>/series/upcoming
Mètode	GET
Paràmetres d'entrada	- team_id: String
Descripció	Retorna les 20 properes sèries de l'equip especificat
Retorn	- upcoming_series: List<Serie>

Ruta	/teams/<team_id>/series/finished
Mètode	GET
Paràmetres d'entrada	- team_id: String - @Path
Descripció	Retorna les 20 últimes sèries finalitzades de l'equip especificat
Retorn	- finished_series: List<Serie>

Ruta	/user/<user_id>/series/upcoming
Mètode	GET

Paràmetres d'entrada	- user_id: String - @Path
Descripció	Retorna les 20 properes sèries dels equips preferits de l'usuari especificat
Retorn	- upcoming_series: List<Serie>

Ruta	/user/<user_id>/series/finished
Mètode	GET
Paràmetres d'entrada	- user_id: String - @Path
Descripció	Retorna les 20 últimes sèries finalitzades dels equips preferits de l'usuari especificat
Retorn	- finished_series: List<Serie>

Ruta	/user/<user_id>/series/upcoming
Mètode	GET
Paràmetres d'entrada	- user_id: String - @Path
Descripció	Retorna les 20 properes sèries dels equips preferits de l'usuari especificat
Retorn	- upcoming_series: List<Serie>

Ruta	/teams/<team_id>/last_results
Mètode	GET
Paràmetres d'entrada	- team_id: String - @Path - date: String - @Query
Descripció	Retorna les últimes 5 sèries de l'equip especificat en la data especificada
Retorn	- last_results: List<Serie>

Ruta	/series/<serie_id>
Mètode	GET
Paràmetres d'entrada	- serie_id: String - @Path
Descripció	Retorna la sèrie especificada
Retorn	- serie: Serie

Ruta	/teams/<team_id>/lineup
Mètode	GET
Paràmetres d'entrada	- team_id: String - @Path - date: String - @Query

Descripció	Retorna l'última alineació de l'equip especificat en la data especificada
Retorn	- lineup: List<JsonElement>

Ruta	/series/<serie_id>/lineups
Mètode	GET
Paràmetres d'entrada	- serie_id: String - @Path
Descripció	Retorna les alineacions de la sèrie especificada
Retorn	- lineups: List<List<JsonElement>>

Ruta	/games_lol/<game_id>
Mètode	GET
Paràmetres d'entrada	- game_id: String - @Path
Descripció	Retorna la partida de LoL especificada
Retorn	- game: Game_lol

Ruta	/games_csgol/<game_id>
Mètode	GET
Paràmetres d'entrada	- game_id: String - @Path
Descripció	Retorna la partida de CSGO especificada
Retorn	- game: Game_csgo

Ruta	/players/<player_id>
Mètode	GET
Paràmetres d'entrada	- player_id: String - @Path
Descripció	Retorna el jugador especificat
Retorn	- player: Player

Ruta	/teams/<team_id>
Mètode	GET
Paràmetres d'entrada	- team_id: String - @Path
Descripció	Retorna l'equip especificat

Retorn	- team: Team
---------------	--------------

Ruta	/users
Mètode	POST
Paràmetres d'entrada	- user: User - @Body
Descripció	Crea l'usuari user
Retorn	- user: User - codi: String

Ruta	/users/<user_id>
Mètode	GET
Paràmetres d'entrada	- user_id: String - @Path
Descripció	Retorna l'usuari especificat
Retorn	- user: User

Ruta	/users/<user_id>
Mètode	POST
Paràmetres d'entrada	- user_id: String - @Path - field: String - @Query - value: String - @Query
Descripció	Actualitza el camp especificat per "field" amb el valor "value" de l'usuari especificat
Retorn	- codi: String - missatge: String

Ruta	/users/<user_id>/fav_teams
Mètode	GET
Paràmetres d'entrada	- user_id: String - @Path
Descripció	Retorna els noms dels equips preferits de l'usuari especificat
Retorn	- fav_teams: List<String>

Ruta	/series/<serie_id>/comments
Mètode	GET
Paràmetres d'entrada	- serie_id: String - @Path - type: String - @Query - order: String - @Query

Descripció	Retorna els comentaris pare de la sèrie especificada (ja siguin els de la prèvia o els del post-partit) amb l'ordre especificat (per puntuació o per data)
Retorn	- comments: List<Comment>

Ruta	/comments
Mètode	POST
Paràmetres d'entrada	- comment: Comment - @Body
Descripció	Crea el comentari especificat a la BD
Retorn	- codi: String - missatge: String

Ruta	/comments/<comment_id>/upvote
Mètode	POST
Paràmetres d'entrada	- comment_id: String - @Path - user_id: String - @Query
Descripció	Realitza l'acció de votar positivament el comentari especificat per l'usuari especificat
Retorn	- codi: String - missatge: String

Ruta	/comments/<comment_id>/downvote
Mètode	POST
Paràmetres d'entrada	- comment_id: String - @Path - user_id: String - @Query
Descripció	Realitza l'acció de votar negativament el comentari especificat per l'usuari especificat
Retorn	- codi: String - missatge: String

Ruta	/comments/<comment_id>
Mètode	DELETE
Paràmetres d'entrada	- comment_id: String - @Path
Descripció	Elimina el comentari especificat
Retorn	- codi: String - missatge: String

Ruta	/comments/<comment_id>/edit
Mètode	POST
Paràmetres d'entrada	- comment_id: String - @Path - user_id: String - @Query

	- text: String - @Query
Descripció	Edita el comentari especificat amb el text especificat
Retorn	- codi: String - missatge: String

Ruta	/comments/<comment_id>/replies
Mètode	GET
Paràmetres d'entrada	- comment_id: String - @Query - order: String - @Path
Descripció	Retorna les rèpliques del comentari especificat en l'ordre especificat (per puntuació o per data)
Retorn	- replies: List<Comments>

8.3. Disseny del front-end

8.3.1. Procés del disseny de la interfície

Els dissenys de la interfície han anat evolucionant a la vegada dels requisits. En l'annex 1 es poden veure els primers dissenys que vaig fer amb paper i boli. Aquests dissenys van passar un procés de canvis proporcionats pels canvis dels requisits i per les opinions dels usuaris esmentats al capítol 7.1 d'aquesta memòria fins a arribar als dissenys finals que s'utilitzen en l'aplicació.

8.3.2. Disseny de la interfície i de la capa de presentació

En aquest apartat es presenten els mockups del disseny de la interfície, les seves versions finals i, per a cada vista, un petit resum de les seves funcionalitats. A més, per a cada vista, s'explica mitjançant diagrames la relació entre els diferents .xml de les vistes i les activitats o els fragments als quals pertanyen, per donar context als següents apartats que parlen sobre el disseny del domini.

Primer, com a introducció al disseny d'interfícies d'aplicacions Android, és oportú conèixer certs termes (totes les definicions han estat extretes de la documentació oficial de desenvolupament d'Android):

- **Vista:** Classe que representa el bloc bàsic de construcció per a components de la interfície d'usuari. [24]
- **Activitat:** Una activitat és una cosa única que l'usuari pot fer. Pràcticament totes les activitats interactuen amb l'usuari, així que la classe activitat s'encarrega de crear una finestra on el desenvolupador pot posar la vista desitjada. [25]
- **Fragment:** Un fragment és una peça de la interfície d'usuari d'una aplicació que pot ser col·locat en una activitat. [26].

Per tal de simplificar, en aquest apartat no s'han inclòs els adaptadors, pel que les vistes tenen una associació <<include>> amb les vistes dels seus fragments.

Pantalla principal

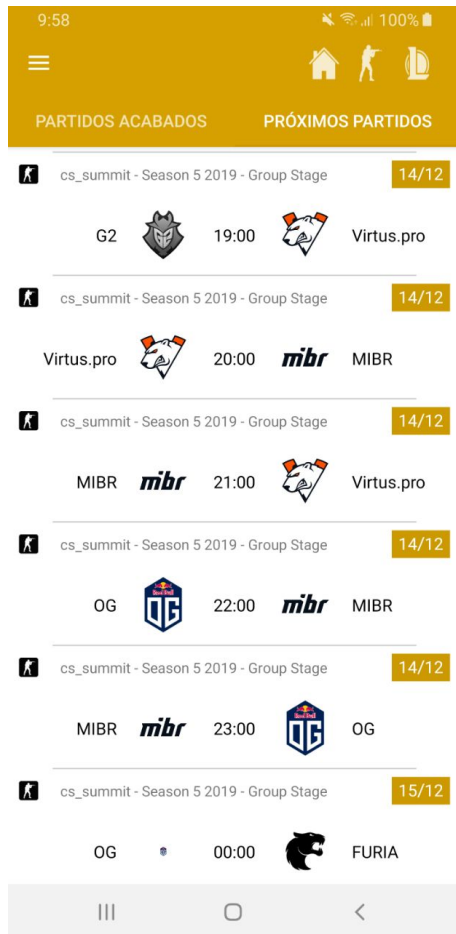


Figura 13. Mockup final de la pestanya de properes sèries de la pantalla principal.
Font: Elaboració pròpia



Figura 14. Mockup final d'un ticket d'una sèrie finalitzada
Font: Elaboració pròpia

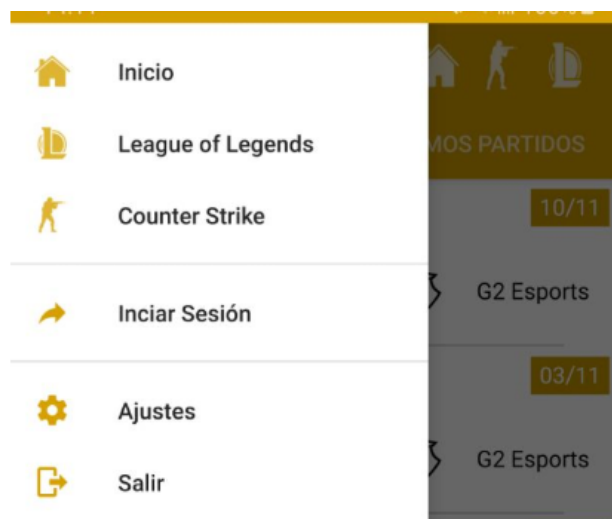


Figura 15. Navigation Drawer.
Font: Elaboració pròpia

Aquesta és la primera pantalla que veu l'usuari quan obre l'aplicació. Consta de dues pestanyes en les quals es llisten les últimes i les properes sèries respectivament. També té un menú lateral (o *navigation drawer*, en termes d'Android) des del que es poden filtrar les sèries per joc, iniciar sessió (o tancar-la), veure el perfil de l'usuari, canviar la configuració de l'aplicació i sortir de l'aplicació.

Aquesta vista, anomenada *activity_main.xml*, està formada per dos fragments, un per a cada pestanya: *fragment_finished_matches.xml* i *fragment_upcoming_matches.xml* i pertany a l'activitat *MainActivity*, mentre que els fragments són gestionats des de *FinishedMatches* i *UpcomingMatches* respectivament. La *MainActivity* s'encarrega de la creació i la gestió dels fragments i el menú lateral, mentre que els fragments la informació bàsica de cada sèrie en una vista anomenada *activity_card_serie.xml* (ticket de la figura 14).

També s'utilitza aquesta vista per mostrar les sèries filtrades o les dels equips preferits de l'usuari.

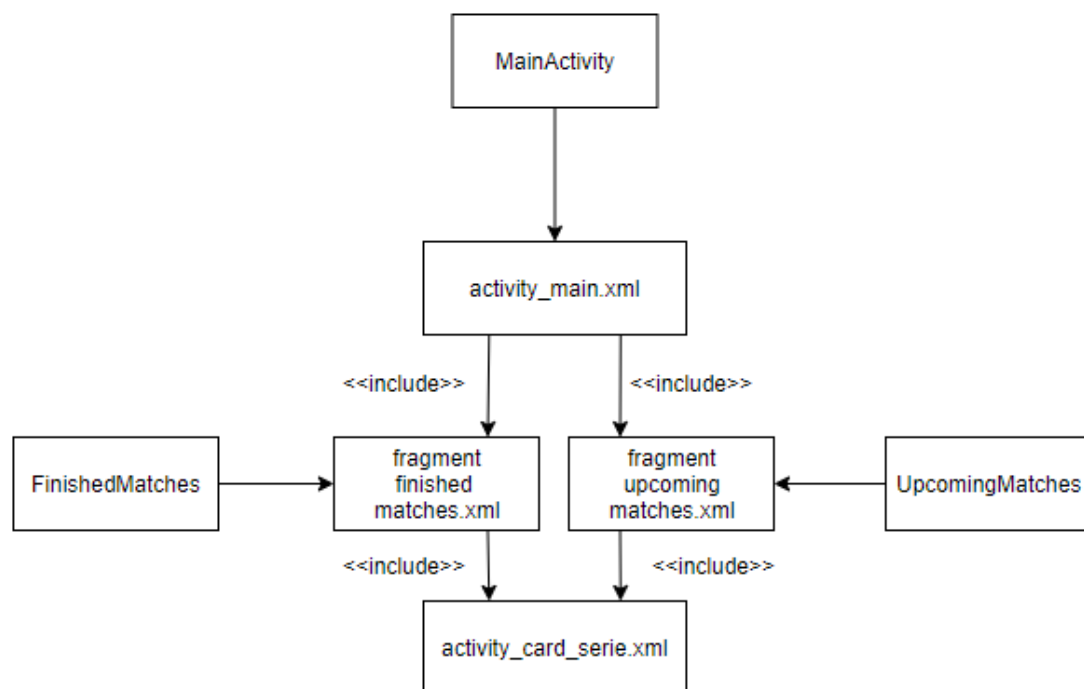
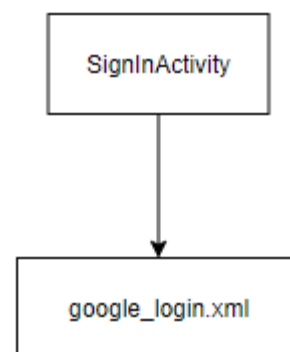
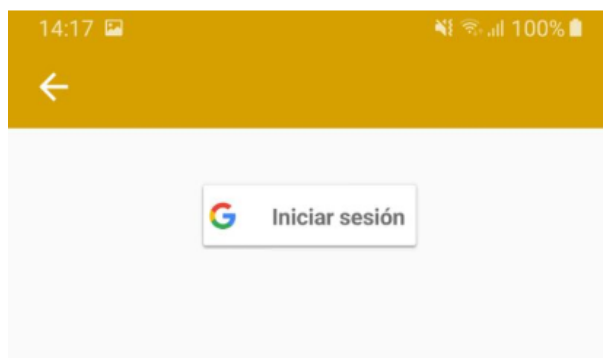


Figura 16. Diagrama de les relacions de la vista principal
Font: Elaboració pròpia

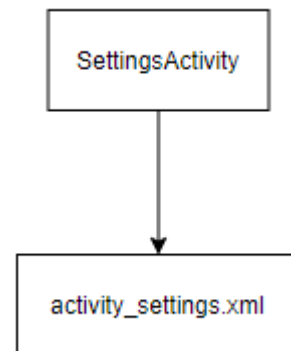
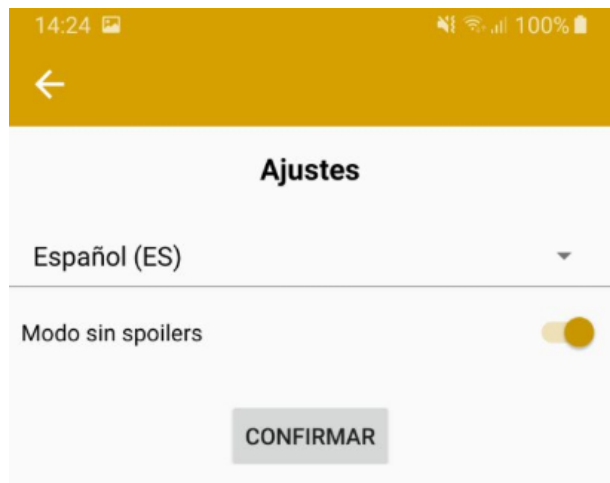
Sign In



Figures 17 i 18. Mockup i diagrama de les relacions de la vista de Sign In
Font: Elaboració pròpia

Vista senzilla des d'on es pot fer el sign in al sistema amb Google. En cas d'èxit, redirigeix a la vista principal.

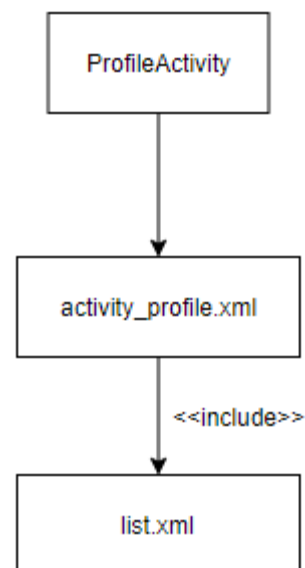
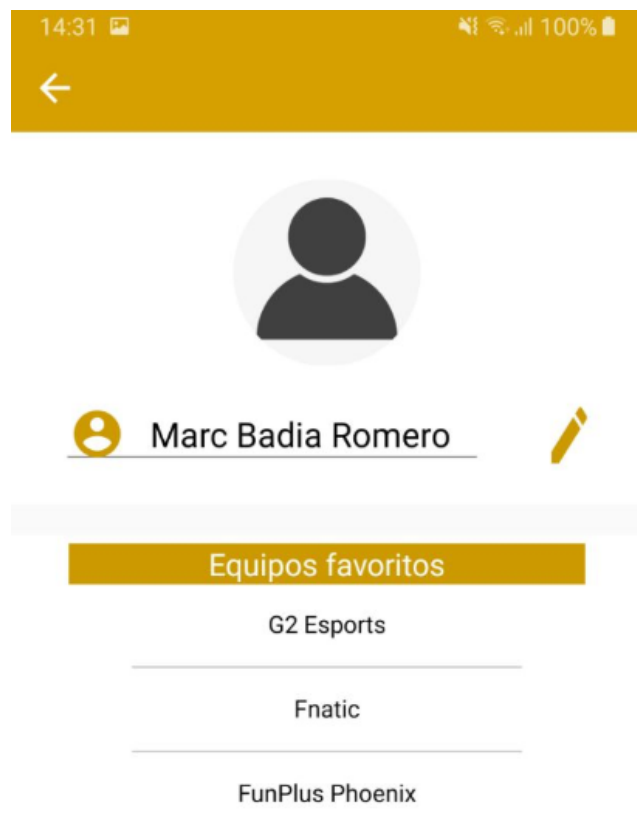
Configuració



Figures 19 i 20. Mockup i diagrama de relacions de la vista de la configuració
Font: Elaboració pròpia

En aquesta vista s'accedeix a la configuració del sistema, des d'on es pot canviar la llengua del sistema i si es volen rebre (o no) spoilers. El botó "Confirmar" guarda els canvis i redirigeix a la vista principal.

Perfil



Figures 21 i 22. Mockup i diagrama de relacions de la vista del perfil
Font: Elaboració pròpia

En aquesta vista es pot veure el perfil de l'usuari que ha iniciat sessió en el sistema. Es permet veure la imatge de perfil i el nom d'usuari, a més d'editar-lo.

Aquesta vista inclou la vista *list.xml*, una vista auxiliar utilitzada per mostrar llistes amb un nombre indefinit d'elements. Cada un dels elements d'aquesta llista redirigeix a la vista de l'equip.

Sèrie

Aquesta és la vista més extensa de l'aplicació, pel que és oportú analitzar-la pestanya a pestanya. Primer, és necessari tenir una visió global de la vista principal de la sèrie.

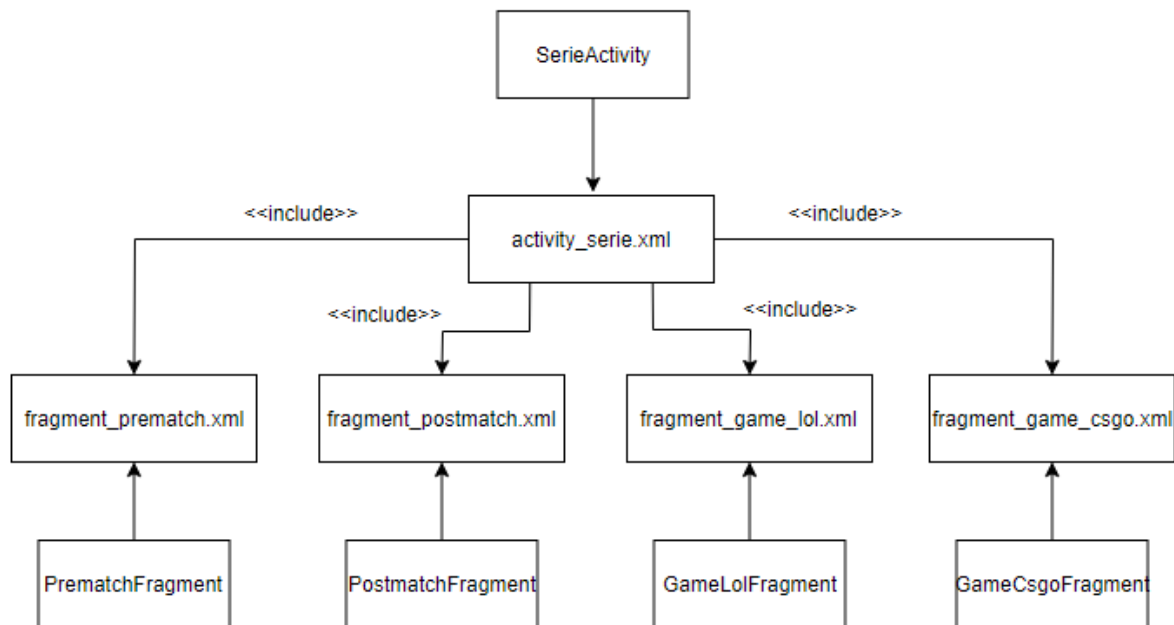


Figura 23. Diagrama de relacions de la vista d'una sèrie
Font: Elaboració pròpia

Com es veu a la figura 23, la vista d'una sèrie (*activity_serie.xml*) està regida per la classe *SerieActivity* i pot tindre fins a quatre tipus de fragments en les seves pestanyes, cada un d'ells amb la seva pròpia classe activitat. Els fragments que tingui una sèrie determinada, depèn de diversos factors: si una sèrie no ha finalitzat, només tindrà una pestanya amb un fragment amb la vista *fragment_prematch.xml*. En canvi, si ha finalitzat tindrà un fragment amb la vista *fragment_postmatch.xml* i tants fragments amb les vistes *fragment_game_lol* o *fragment_game_csgo* (depenent de si és una sèrie de LoL o de CSGO) com partides hi hagi en la sèrie.

Ara analitzem més en profunditat cada una de les vistes d'aquests fragments.

Pre-partit

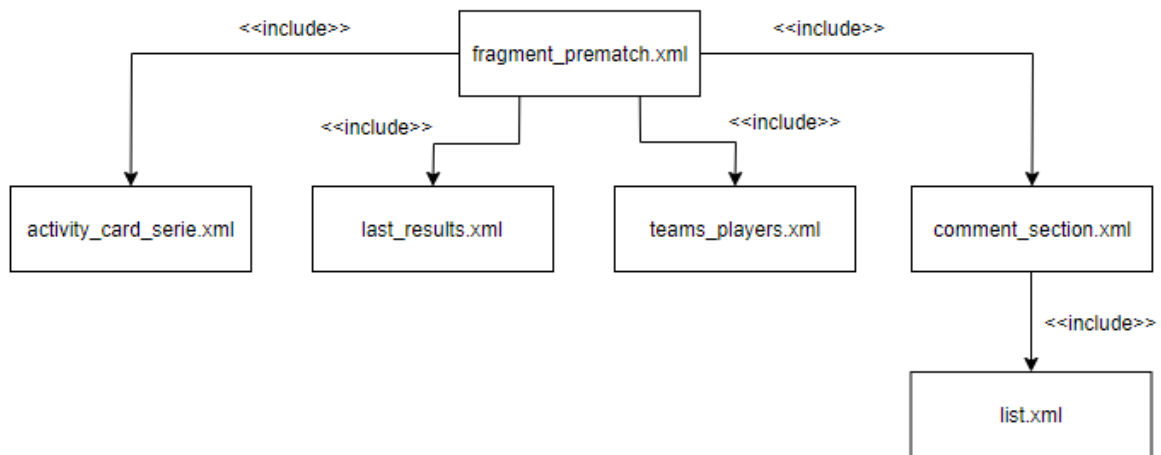
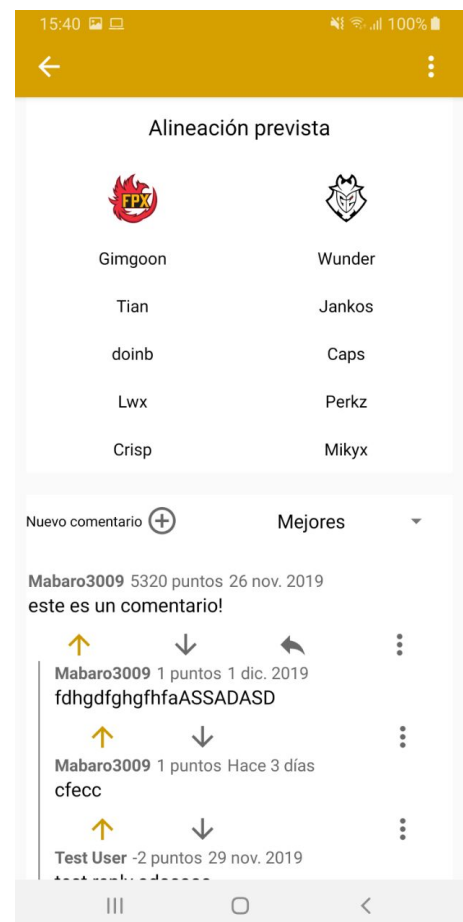


Figura 24. Diagrama de relacions de la vista del fragment PrematchFragment
Font: Elaboració pròpia



Figures 25 i 26. Mockups de la vista del fragment PrematchFragment
Font: Elaboració pròpia

En aquest fragment hi ha diferents components. Primer, tenim el ticket de la sèrie en la vista de *activity_card_serie.xml*, igual que en les vistes dels fragments de properes i anteriors sèries.

Després, apareix una llista de les últimes sèries jugades per cadascun dels equips, que dirigeixen a la vista de la sèrie especificada. També tenim l'alineació prevista dels equips i la llista dels comentaris. Aquesta llista de comentaris també utilitza la vista auxiliar *list.xml* explicada anteriorment i prement "Nou comentari", apareixerà una vista popup anomenada *popup_new_comment.xml* on es pot escriure un nou comentari.

Post-partit

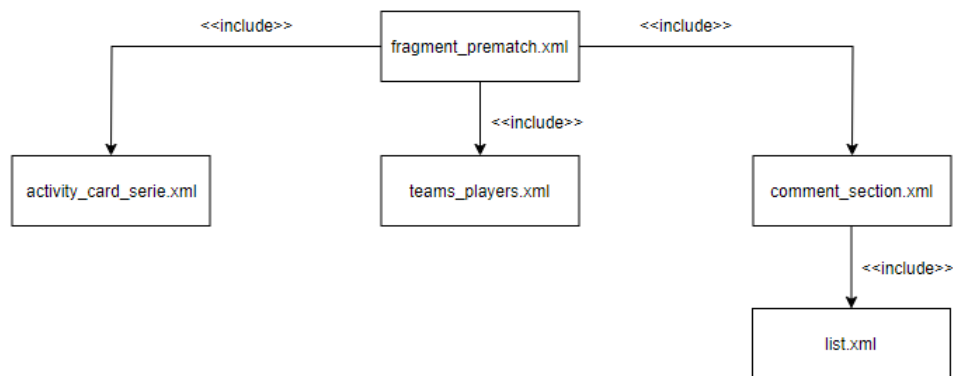
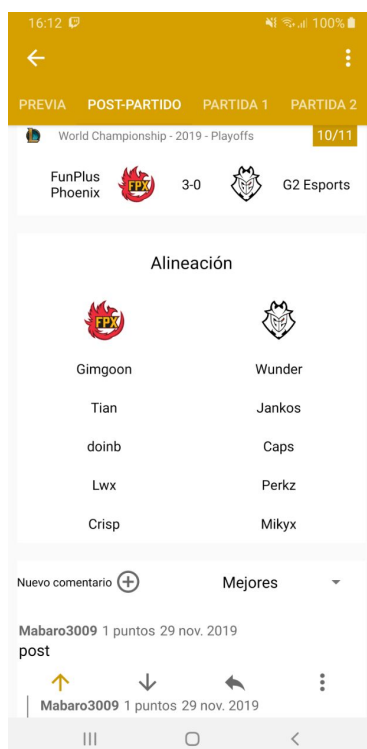


Figura 26. Diagrama de relacions de la vista del fragment PostmatchFragment

Font: Elaboració pròpia



Aquesta vista és similar a l'anterior, amb la diferència de què es mostra el resultat en comptes de l'hora en el ticket, l'alineació titular real que va jugar la sèrie i els comentaris són exclusius al postpartit.

Figura 27. Mockup del post-partit

Font: Elaboració pròpia

Game LoL

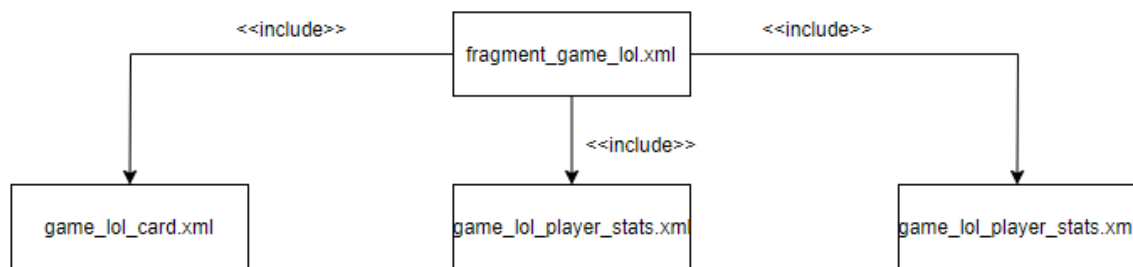


Figura 28. Diagrama de relacions de la vista del fragment GameLoLFragment
Font: Elaboració pròpia



Figura 29. Mockup de la vista d'una partida de LoL
Font: Elaboració pròpia

En aquesta vista exclusiva de les sèries de LoL, tenim diverses vistes auxiliars.

Primer, en *game_lol_card*, es mostra informació bàsica sobre la partida, com el temps de duració, el guanyador de la partida, estadístiques dels equips i els campions que ha bloquejat cada equip.

Després tenim una vista *game_lol_stats.xml* per a cada equip. En aquesta vista es mostren les estadístiques de cada jugador en aquesta partida en format de taula. Per a cada jugador s'especifica:

- Nom del jugador.
- Imatge del campió que ha jugat.
- El seu K/D/A.
- Els seus cs.
- L'or que ha aconseguit (en milers).

Game CSGO

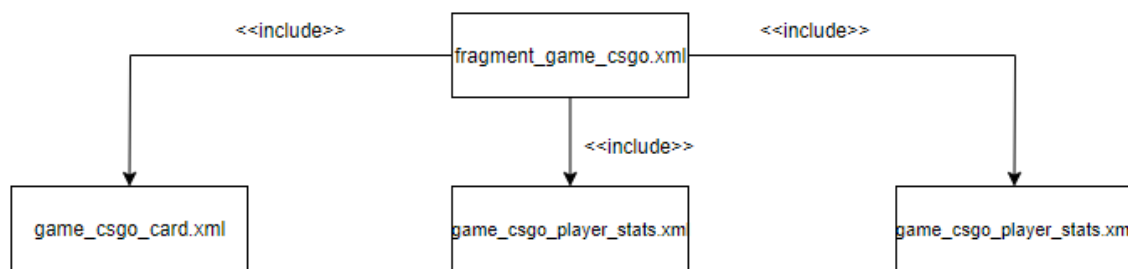


Figura 30. Diagrama de relacions de la vista del fragment GameCsgoFragment

Font: Elaboració pròpia



Figura 31. Mockup de la vista d'una partida de CSGO

Font: Elaboració pròpia

Variant de la vista anterior per a partides de CSGO.

Com es pot comprovar en la figura 30, aquesta vista internament és igual a la de LoL.

Els canvis són que en la vista auxiliar `game_csgo_card` hi ha el nombre de rondes guanyades per a cada equip a més de la imatge i el nom del mapa en el qual s'ha jugat la partida.

Pel que fa a les `game_csgo_player_stats.xml`, per a cada jugador es mostra:

- Nom del jugador.
- K/D del jugador.
- ADR del jugador.

Jugador i Equip

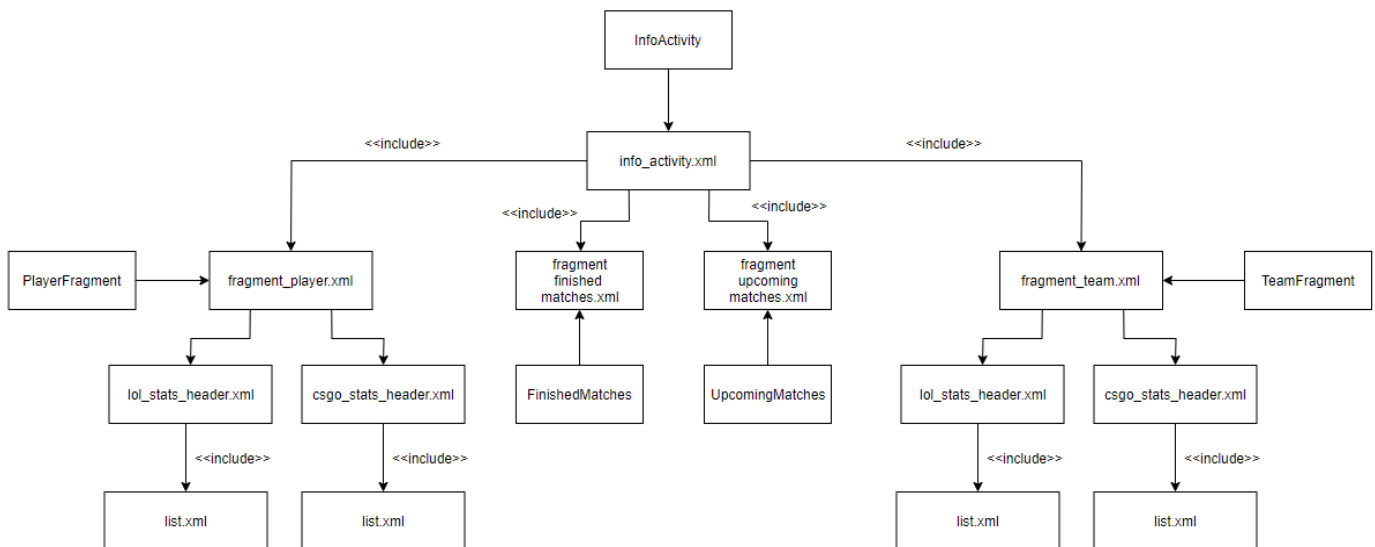
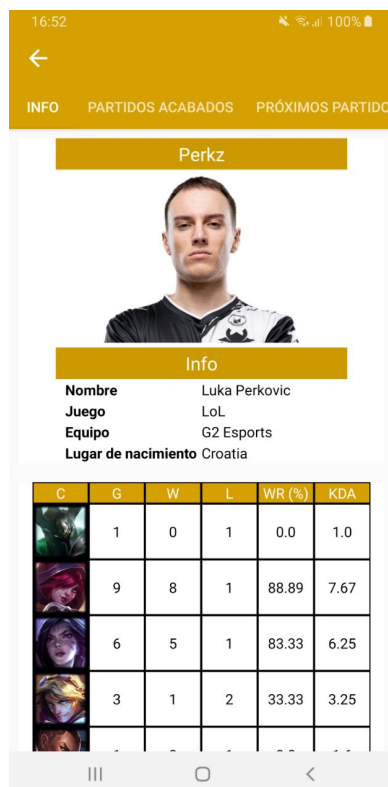


Figura 32. Diagrama de relacions de la vista del fragment d'informació
Font: Elaboració pròpia

Aquesta vista *info_activity.xml* és una mica especial en el sentit que s'utilitza tant per a jugador com per a equips ja que hi ha molt poques diferències entre la informació mostrada entre les dues classes, tal com es pot veure en la figura 32.



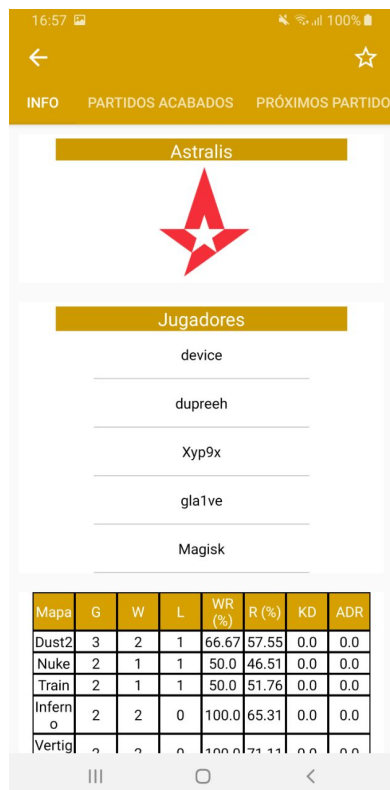
En una vista de jugador, veiem la següent informació del jugador:

- Nom del jugador.
- Imatge del jugador.
- Nom real del jugador.
- Joc al que juga professionalment el jugador.
- Equip del jugador.
- Lloc de naixement del jugador.

A més, es mostren estadístiques dels campsions del jugador si és jugador de LoL o estadístiques del jugador en els diferents mapes si és jugador de CSGO.

També es mostren en altres pestanyes les vistes *fragment_finished_matches.xml* i *fragment_upcoming_matches.xml*, on es mostren les sèries finalitzades i properes del jugador, igual que en la vista principal.

Figura 33. Mockup de la vista d'un jugador de LoL
Font: Elaboració pròpia



En la vista d'un equip, veiem la següent informació:

- Nom de l'equip.
- Imatge de l'equip.
- Llista dels jugadors de l'equip.

Igual que en les vistes dels jugadors, es mostren estadístiques dels campsions de l'equip si és un equip de LoL o estadístiques de l'equip en els diferents mapes si és un equip de CSGO.

També es mostren en altres pestanyes les vistes *fragment_finished_matches.xml* i *fragment_upcoming_matches.xml*, on es mostren les sèries finalitzades i properes de l'equip, igual que en la vista principal.

Un equip pot ser marcat (o desmarcat) com a preferit per l'usuari prement la icona en forma d'estrella.

Figura 33. Mockup de la vista d'un equip de CSGO
Font: Elaboració pròpia

8.3.3. Mapa navegacional

En la figura 34 es mostra el mapa navegacional d'àmbit sistema de l'aplicació obviant les vistes auxiliars.

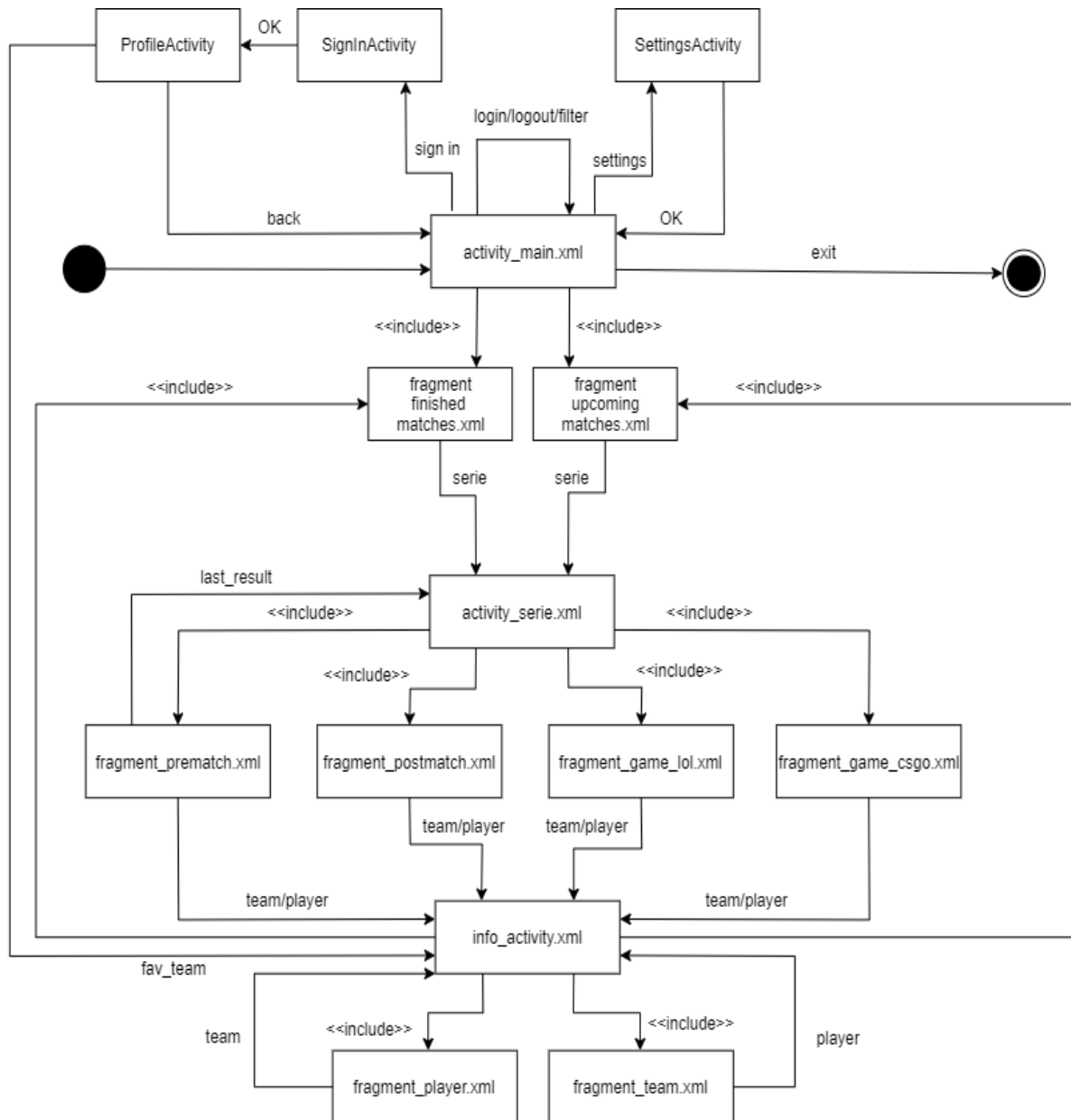


Figura 34. Mapa navegacional d'àmbit Sistema
Font: Elaboració pròpia

8.3.4. Diagrama de classes del disseny

Primer, cal tenir una visió general de les relacions entre els diferents objectes de l'aplicació.

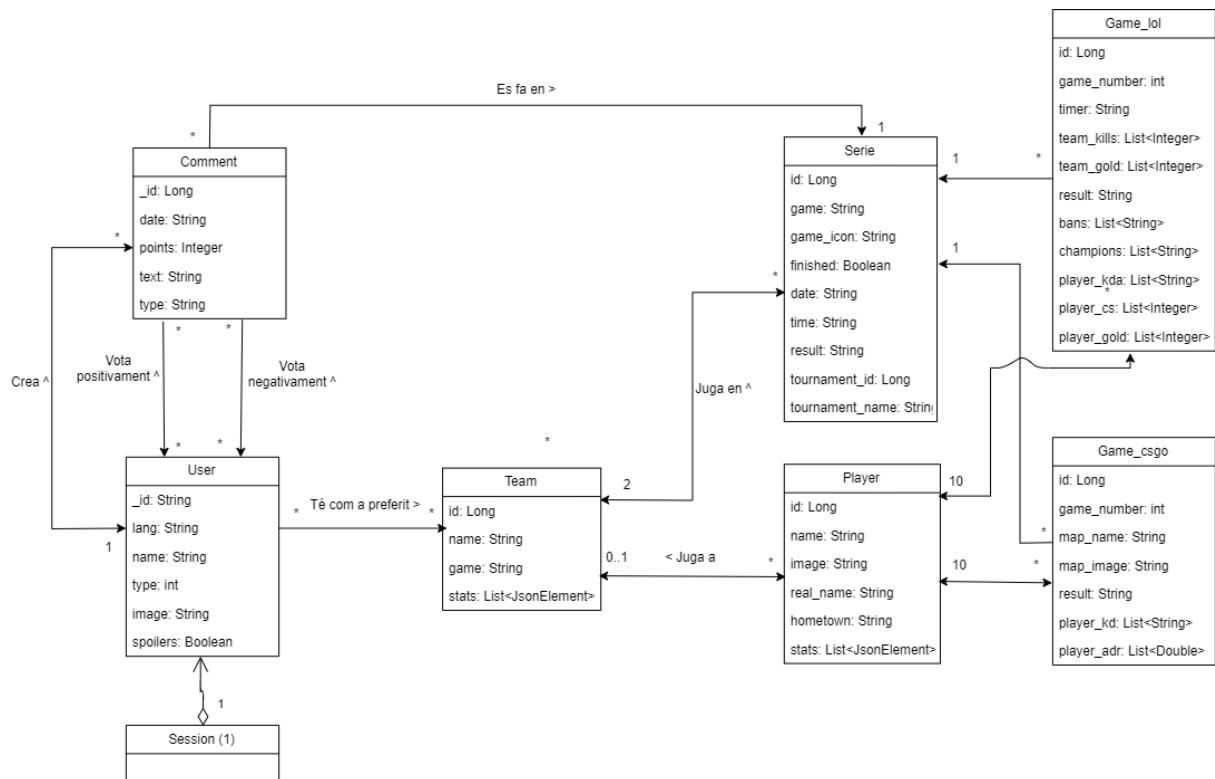


Figura 35. Model conceptual de les dades del front-end

Font: Elaboració pròpia

Com s'ha explicat anteriorment, en aquest projecte primer s'ha fet el disseny del back-end, pel que no és cap sorpresa que el diagrama de la figura 35 sigui una simplificació del diagrama lògic de la base de dades de la figura 12. De fet, aquest disseny s'ha fet tenint en compte que els diferents objectes podran ser extrets de la base de dades a plaer, pel que no és necessari incloure certes classes com poden ser "Resource" o "Tournament", ja que no tenen la suficient rellevància dins de l'aplicació i els seus camps importants poden ser inclosos en altres classes, en aquest cas "Game_lol" i "Game_csgo" per "Resource", i "Serie" per "Tournament".

Per tal d'aclarir conceptes, aquest model conceptual no implica cap base de dades interna a l'aplicació del mòbil, totes les dades s'extreuen de la base de dades del back-end. L'esquema és simplement una representació visual de les relacions entre els objectes del sistema.

A més, s'ha inclòs una classe singleton "Session" que ja estava present en el model conceptual de les dades (figura 10), però no era necessària en el diagrama lògic de la base de dades (figura 12). Aquesta classe representa l'usuari amb el què s'ha iniciat sessió en el sistema.

Ara veurem les relacions entre les classes de la figura 35, les activitats i fragments presentats a l'anterior apartat i els diferents controladors de presentació i domini, serveis i adaptadors necessaris per el correcte funcionament de l'aplicació.

En comptes de mostrar un sol diagrama gran amb totes les relacions, es mostren i s'expliquen diversos diagrames més petits per tal de facilitar la lectura del document.

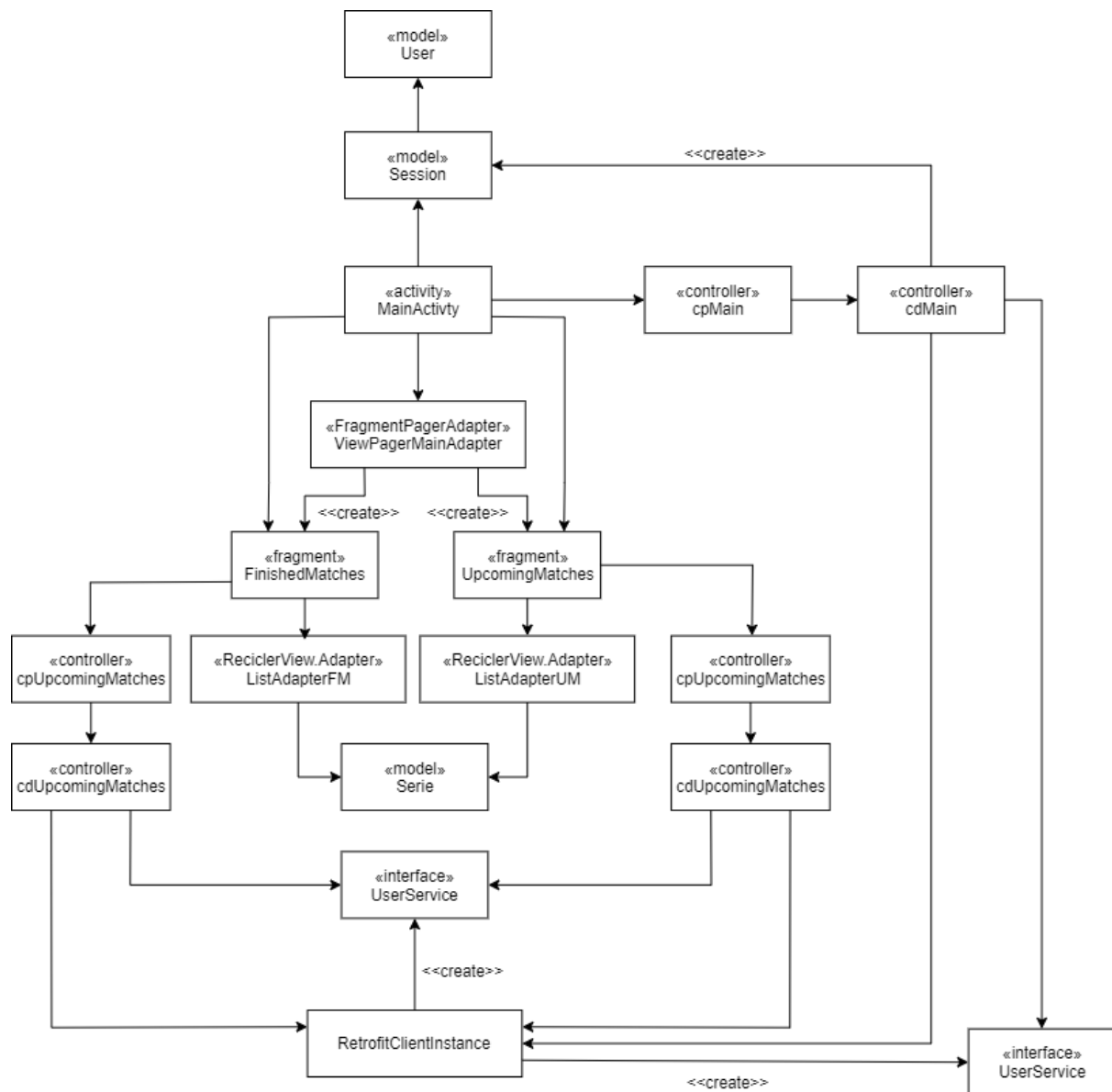


Figura 36. Diagrama de les classes relacionades amb MainActivity
Font: Elaboració pròpia

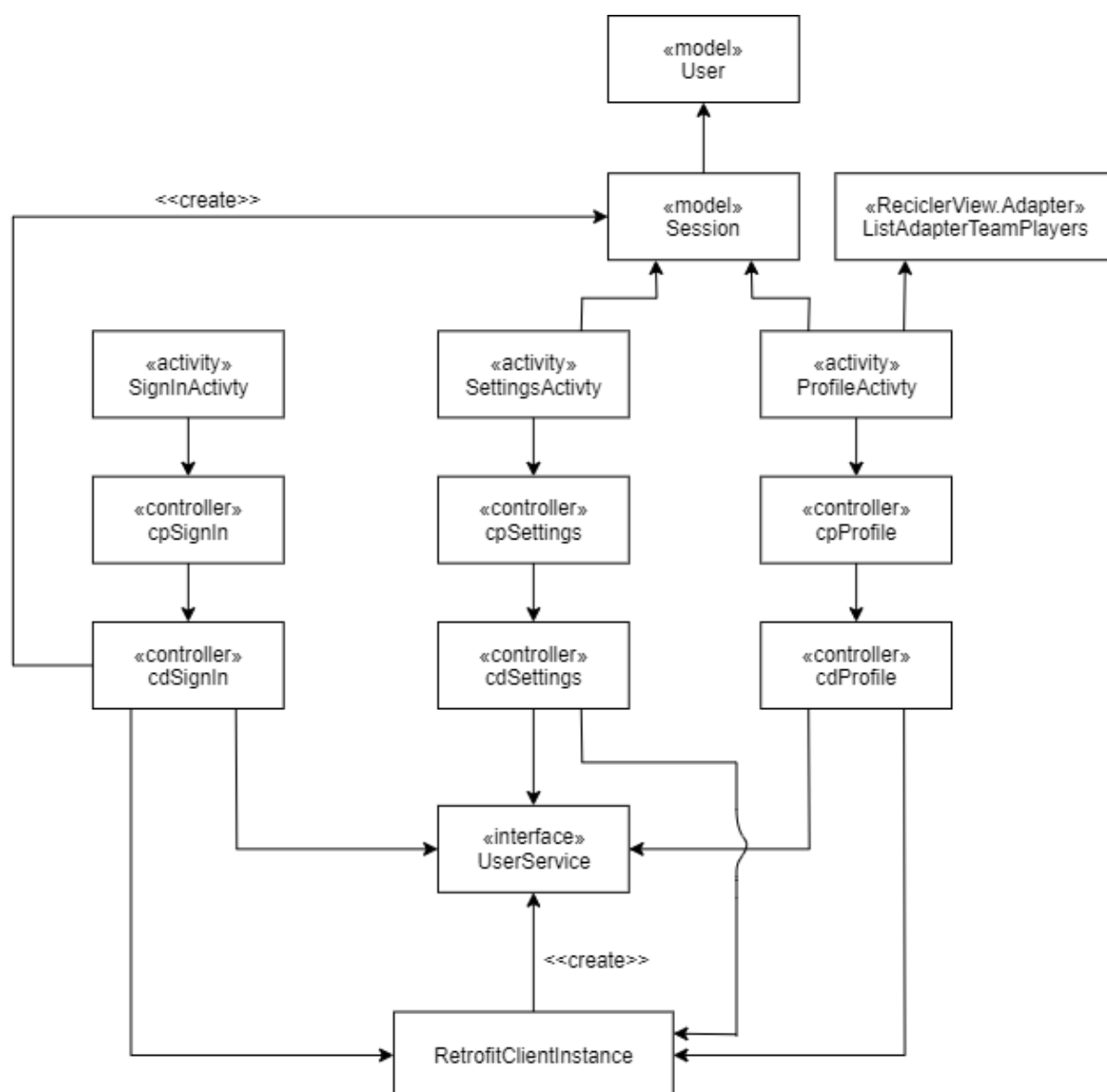


Figura 37. Diagrama de les classes relacionades amb SignInActivity, SettingsActivity i ProfileActivity
Font: Elaboració pròpia

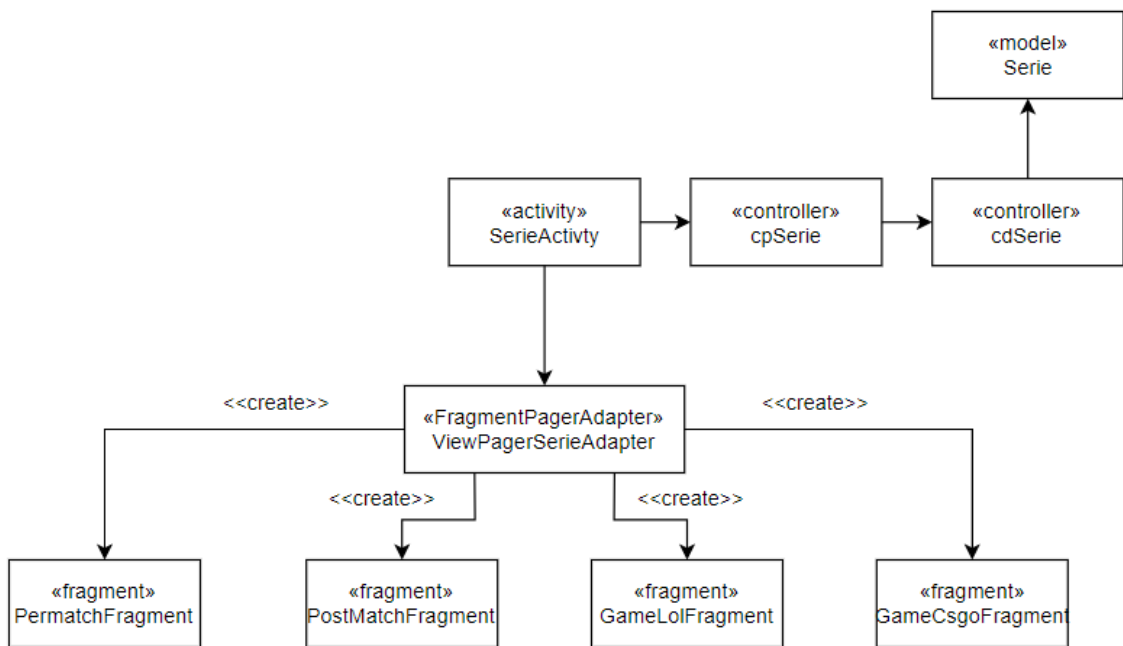


Figura 38. Diagrama de les classes relacionades amb SerieActivity
Font: Elaboració pròpia

Igual que en la figura 23, en la figura 38 només es mostren les classes directament relacionades amb SerieActivity. En les figures 39 a 41 es mostren les relacionades amb cadascun dels fragments.

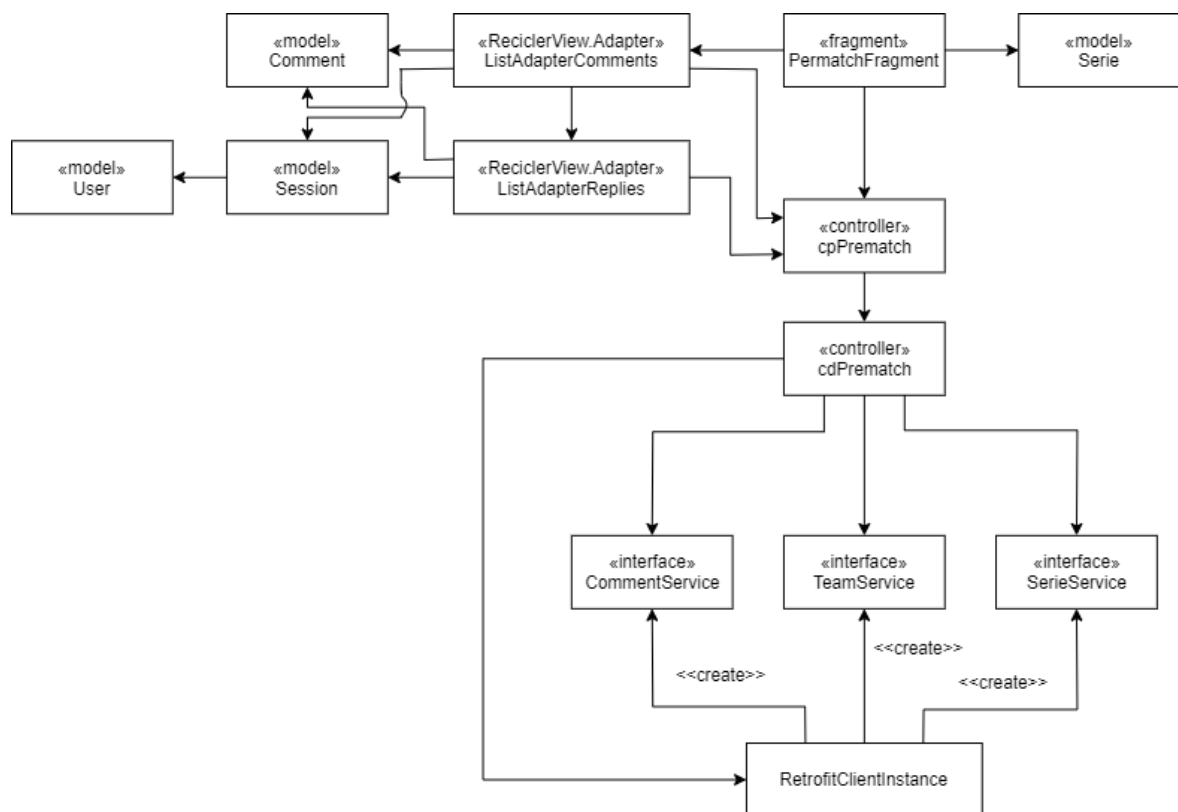


Figura 39. Diagrama de les classes relacionades amb PrematchFragment
Font: Elaboració pròpia

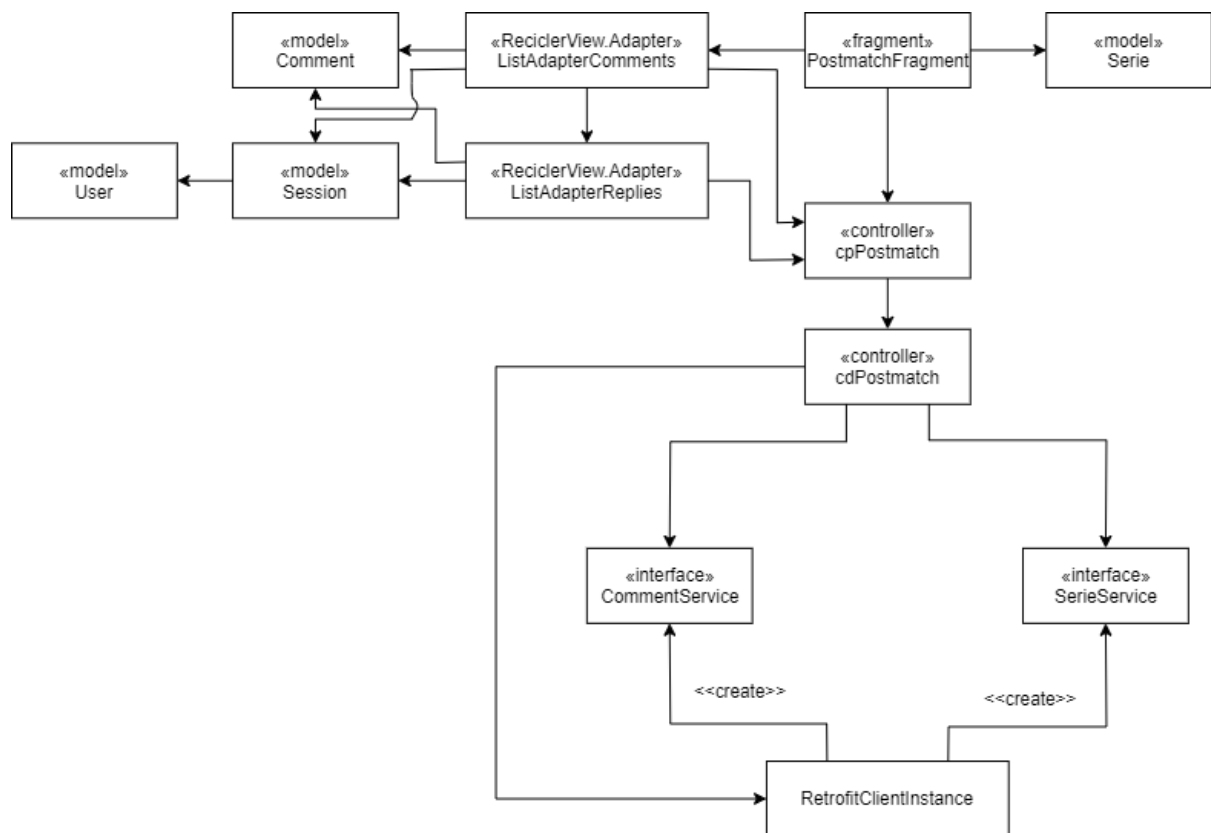


Figura 40. Diagrama de les classes relacionades amb PostmatchFragment
Font: Elaboració pròpia

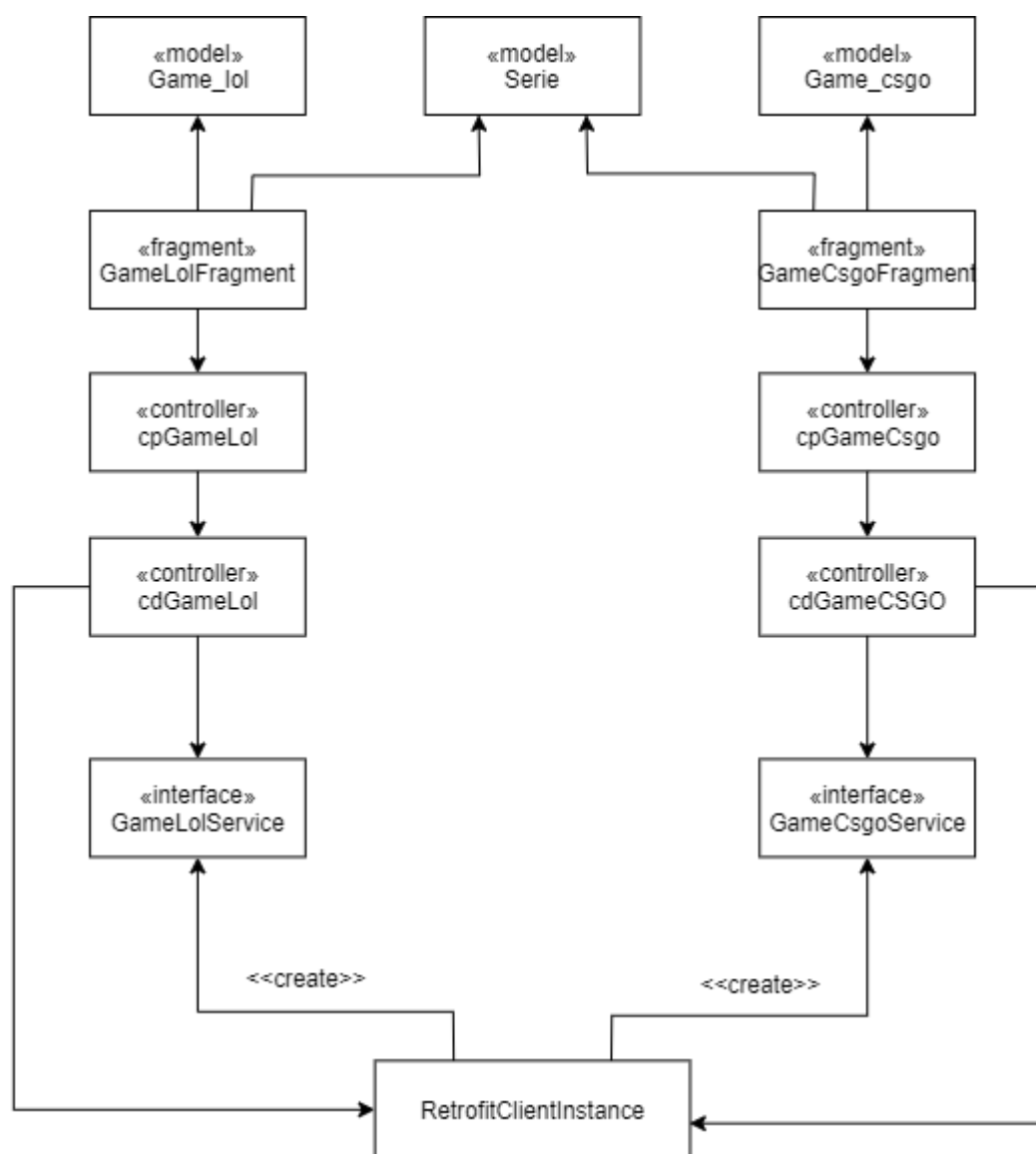


Figura 41. Diagrama de les classes relacionades amb GameLolFragment i GameCsgoFragment
Font: Elaboració pròpia

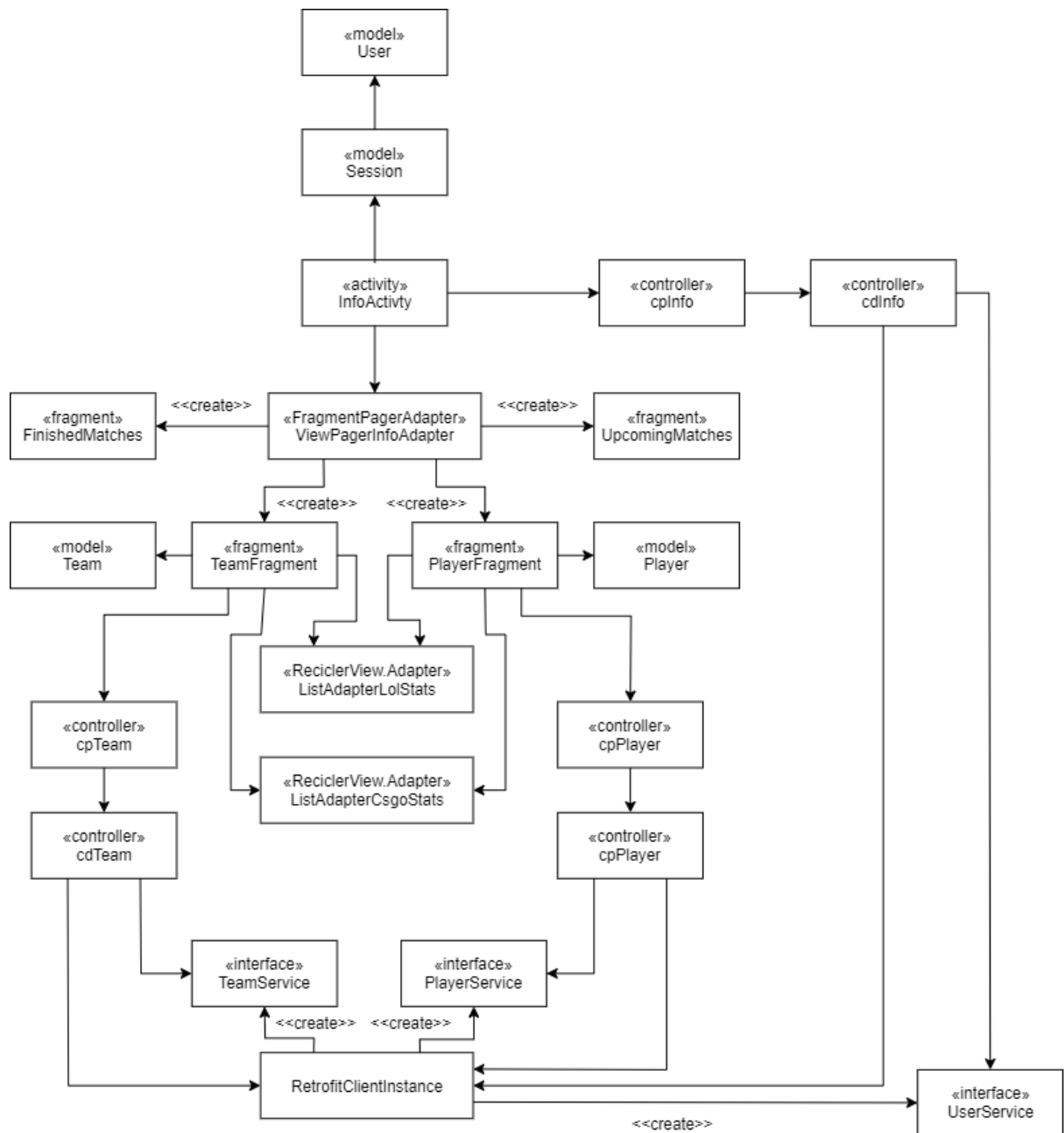


Figura 42. Diagrama de les classes relacionades amb InfoActivity
Font: Elaboració pròpia

En la figura 42 s'han ignorat les classes relacionades amb *FinishedMatches* i *UpcomingMatches* perquè ja es mostren en la figura 36.

8.4. Patrons utilitzats

8.4.1. Patró Model-Vista-Controlador

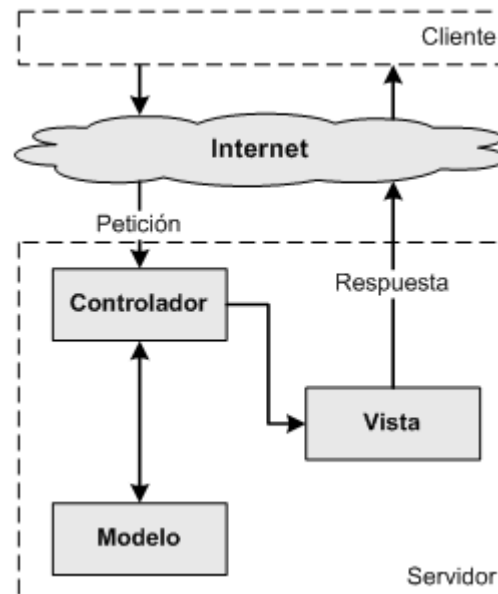


Figura 43. Patró Model-Vista-Controlador

Font: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122>

Per l'aplicació mòbil s'ha utilitzat el patró Model-Vista-Controlador. Com es veu en la figura 43, aquest patró tracta de separar la interfície gràfica del codi de funcionament d'una aplicació per tal que el codi estigui millor estructurat i sigui més fàcil trobar i solucionar els errors. A més facilita l'actualització i la reutilització del codi. [27]

Aquest patró està compost de tres components. [28]

- **Model.** El model representa la informació del món real que s'ha de veure reflectida en el sistema i es comunica amb la base de dades.
- **Controlador.** El controlador rep les peticions de l'usuari, actua sobre el model i comunica els resultats a la vista.
- **Vista.** Les vistes mostren el contingut dels models tractat pels controladors a l'usuari.

8.4.2. Patró singleton

Com es pot veure en la figura 35, la classe "Session" de l'aplicació mòbil és una classe singleton.

El patró singleton garanteix que una classe té una única instància, a la vegada que proveeix un punt d'accés global a aquesta instància. [29]

8.5. Exemples de diagrames de registre

No té sentit fer tots els diagrames de registre del sistema, ja que n'hi ha molts que són molt similars. Per aquest motiu, en aquesta secció es mostren els diagrames més representatius que mostren la interacció entre les classes del sistema.

Molts dels diagrames del sistema s'assemblen als mostrats en aquesta secció.

8.5.1. Sign In i registre

En la figura 44 es veu com a l'activitat li arriba un input de l'usuari amb el compte de Google amb què vol iniciar sessió. Es passa aquesta informació pels controladors, fins que finalment el controlador de domini crea l'*UserService* a través del *RetrofitClientInstance*.

Es pregunta a la interfície per l'usuari que correspon a l'id proporcionat pel compte de Google i es retorna l'usuari de la base de dades. En cas que sigui null significa que és la primera vegada que s'intenta accedir al sistema amb aquesta conta de Google, pel que cal registrar-la a la base de dades.

Finalment, es crea la sessió de l'usuari i es canvia la llengua de l'aplicació per la corresponent.

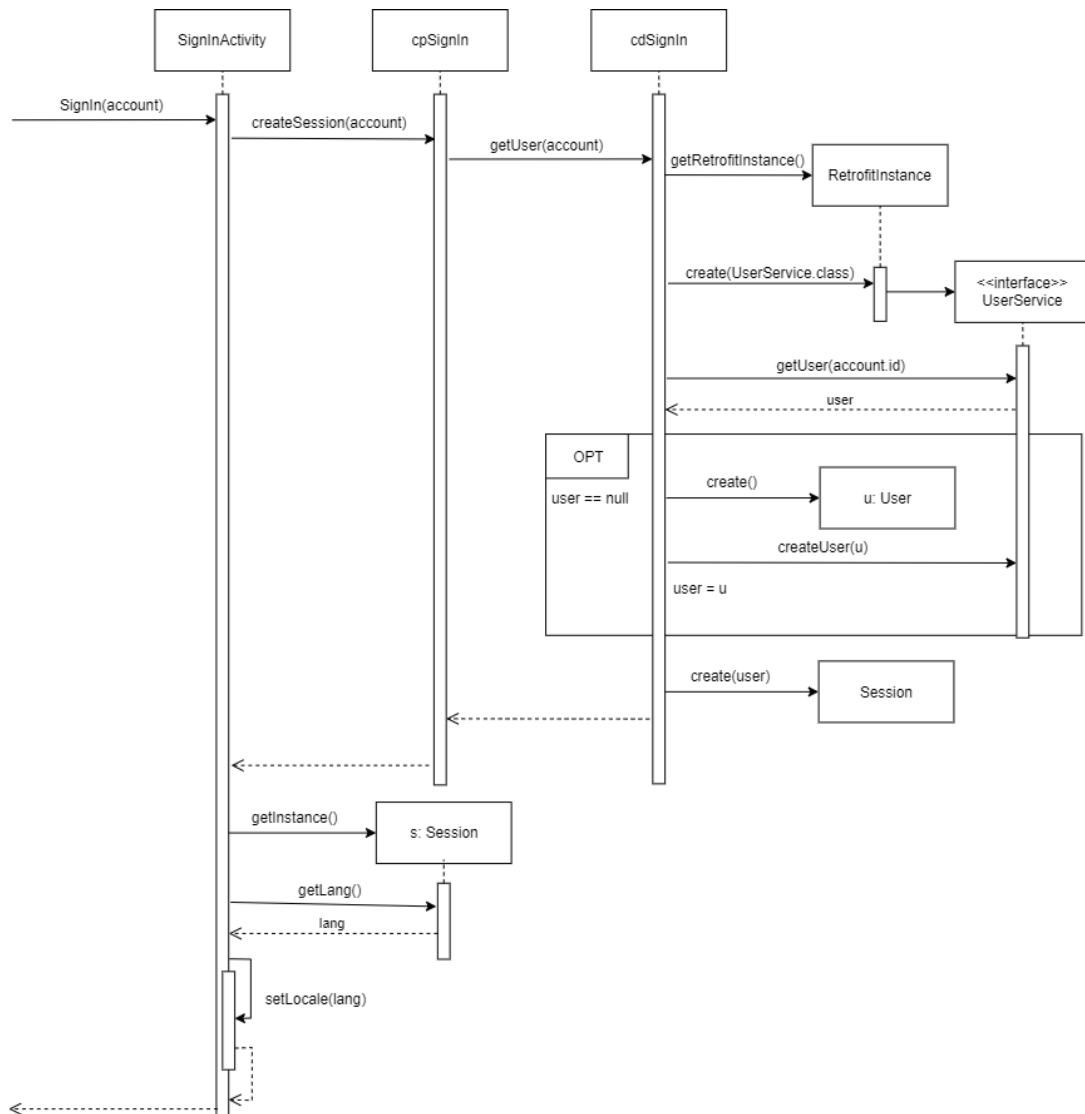


Figura 44. Diagrama de seqüència del sign in i el registre
Font: Elaboració pròpia

8.5.2. Canviar llengua

En la figura 45 es mostra el diagrama de seqüència per canviar la llengua d'un usuari que ha iniciat sessió en el sistema.

El diagrama comença quan l'usuari prem el botó de canviar la llengua amb una llengua seleccionada. Primer, es canvia el Locale de l'aplicació per tal que es tradueixi l'aplicació i, després, s'actualitza l'usuari a la base de dades.

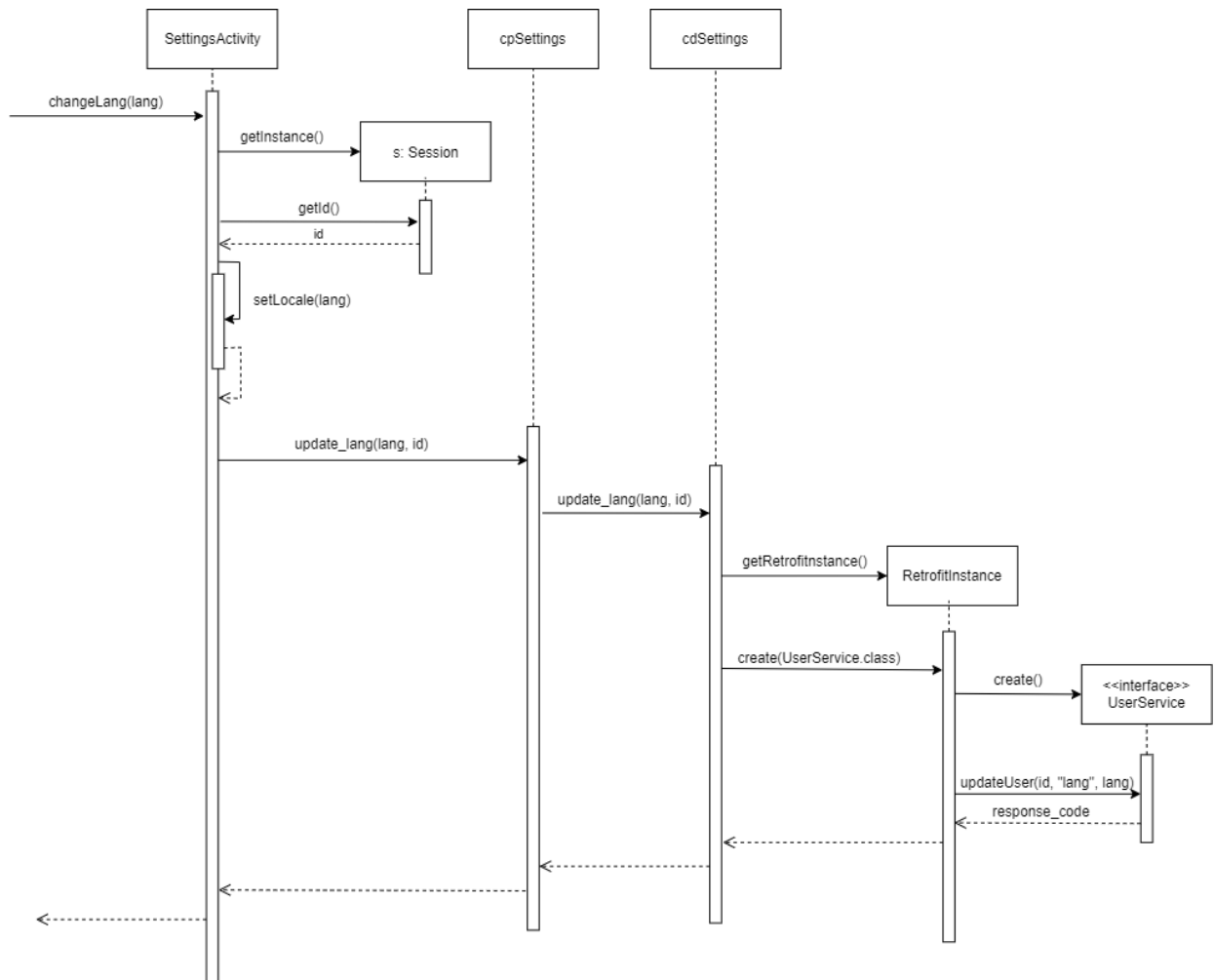


Figura 45. Diagrama de seqüència per canviar la llengua

Font: Elaboració pròpia

8.5.3. Editar nom d'usuari

En el cas d'ús de la figura 46, l'usuari indica el seu nou nom d'usuari. Si es comprova que el nou nom no compleix certs requisits, s'indica a través d'un error. En cas contrari s'actualitza la base de dades amb el nou nom.

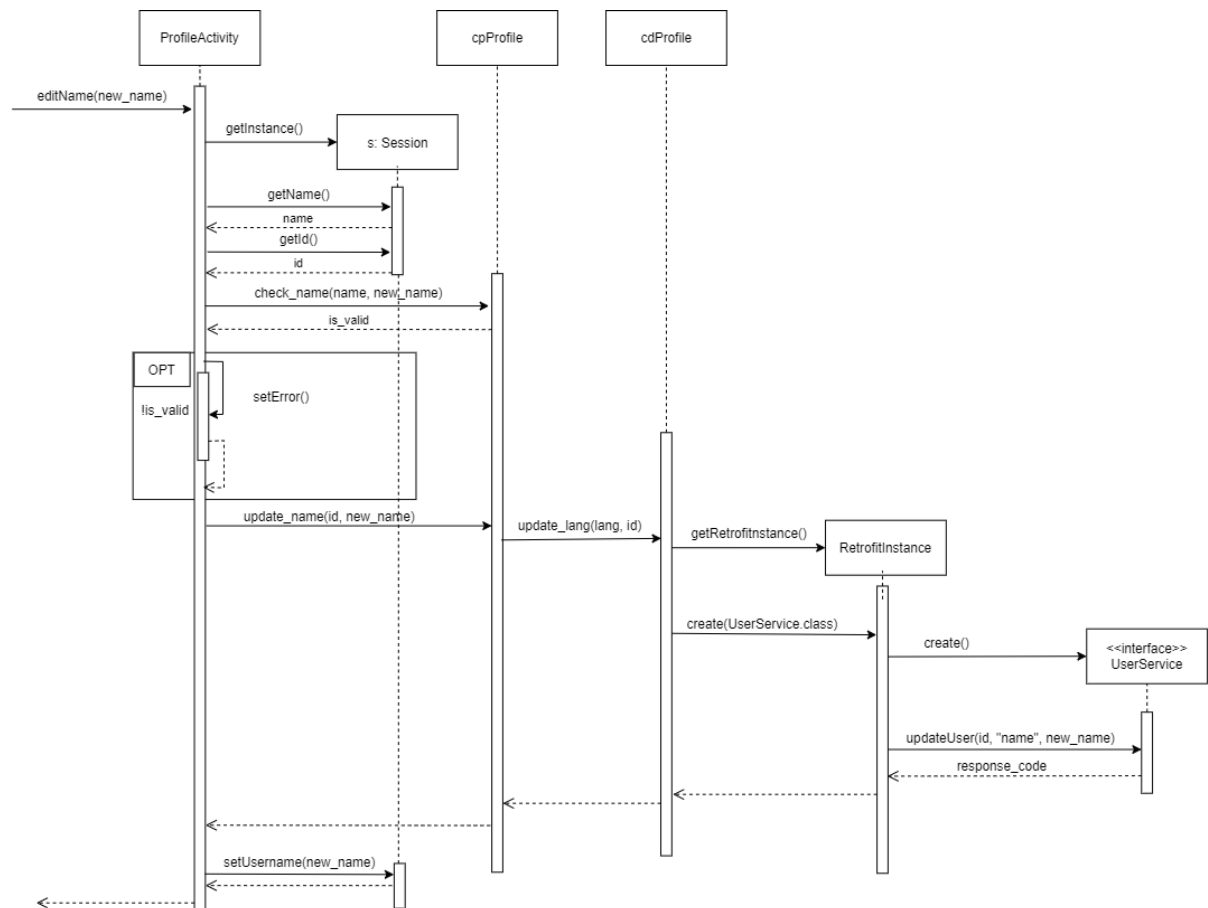


Figura 46. Diagrama de seqüència per editar el nom d'usuari

Font: Elaboració pròpia

8.5.4. Carregar properes sèries

En aquest diagrama (figura 47), es mostra com es carreguen les properes partides sense filtres.

Primer, l'usuari indica que vol veure les properes partides (o obre l'aplicació per primera vegada), pel que la funció *loadFeed* té els valors per defecte "game" i "all", que indiquen que es volen veure totes les sèries dels jocs. En el controlador de presentació es controla el tag i es crida a la funció correcta del controlador de domini, que carrega les sèries demanades.

Una vegada es tenen les sèries, es crea l'adaptador i es posa en la RecyclerView per tal de poder mostrar-les.

Com s'ha dit, el diagrama de la figura 47 mostra l'exemple de les properes sèries sense filtres (és a dir, dels dos jocs), però l'esquelet del diagrama és el mateix amb altres filtres, simplement canviant les funcions que es criden.

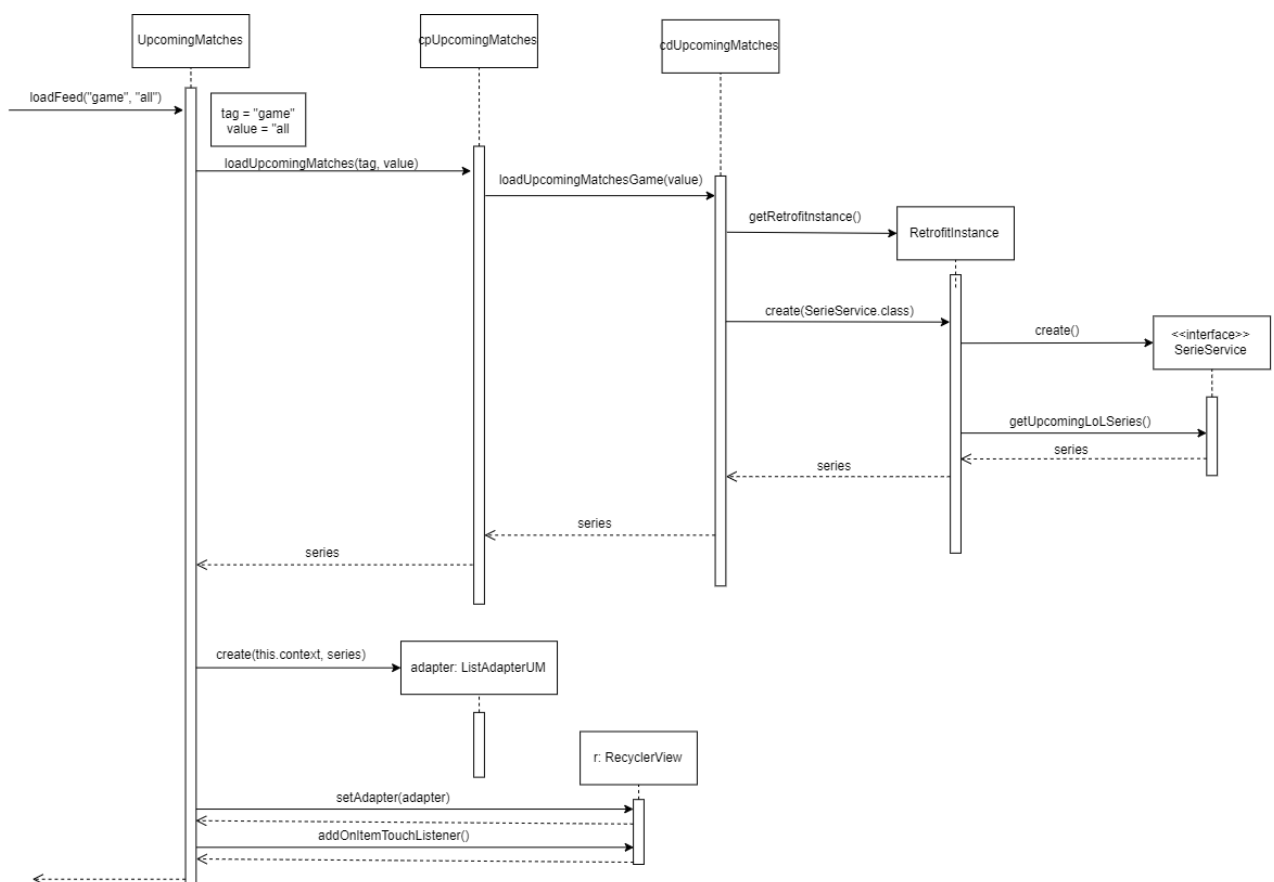


Figura 47. Diagrama de seqüència per carregar les properes sèries

Font: Elaboració pròpia

9. Implementació

Ara que tenim una visió del funcionament intern del sistema, cal veure com s'ha implementat.

En aquest capítol s'expliquen i es justifiquen les diferents tecnologies, llibreries i llenguatges emprats (secció 9.1), així com les eines de desenvolupament i altres serveis utilitzats (secció 9.2). Al final del capítol, s'exemplifiquen els aspectes més rellevants del codi de la implementació (secció 9.3).

9.1. Tecnologies, llibreries i llenguatges emprats

Primer, en aquest apartat, es repassen les tecnologies, llibreries i llenguatges utilitzats en el desenvolupament tant de l'aplicació mòbil com del back-end del sistema.

9.1.1. Python

Python [30] és un "llenguatge de programació que et permet treballar més de pressa i integrar els teus sistemes de forma més efectiva", segons la pàgina web oficial [31].



Figura 48. Logo de Python
Font: www.python.org

En aquest sistema, Python s'ha utilitzat per programar els dos scripts que mantenen actualitzada la base de dades (un per afegir informació de l'API externa i l'altre per afegir informació que no està disponible a l'API externa, com les estadístiques de les partides) i per fer l'API que comunica el front-end amb el back-end. S'ha utilitzat la versió 3.7, ja que era la més recent en el començament del projecte [32].

Les llibreries més importants que s'han utilitzat són:

- **pandascore** [33]. Aquesta és la llibreria oficial de l'API de PandaScore [21], que permet accedir a ella d'una manera ràpida i senzilla. Retorna sempre un diccionari de la resposta en JSON.
- **pymongo** [34]. PyMongo és la llibreria recomanada per treballar amb MongoDB en Python.
- **selenium** [35]. La llibreria de Selium permet automatitzar navegadors web. És especialment útil per fer la pràctica del *HTML Scraping*, que s'explica més endavant.
- **urllib** [36]. Aquesta llibreria permet tractar URLs. En el sistema s'utilitza per llegir JSONs de *paths* de l'API de PandaScore que no estan disponibles mitjançant la seva llibreria.

- **PIL [37].** *Python Imaging Library*. Com el nom indica, és una llibreria que permet tractar imatges. S'utilitza per detectar si una imatge és vàlida o no (si dóna una excepció al obrir la imatge, no és vàlida).
- **lxml [38].** Aquesta llibreria s'utilitza per tractar XML i HTML en Python. Igual que la llibreria de selenium, s'utilitza per fer *HTML Scraping*.
- **Flask.** Segons la seva pàgina web, Flask és un “framework lleuger d'aplicació web WSGI” [39]. En aquest sistema s'ha utilitzat per fer l'API per comunicar el front-end amb el back-end.

9.1.2. MongoDB

Tal com s'ha justificat en apartats anteriors, la base de dades que s'utilitza en aquest projecte és una base de dades MongoDB [19] que es modifica i actualitza gràcies a dos scripts fets amb Python. A més, una API Flask s'encarrega de comunicar aquesta base de dades amb el front-end del sistema.



Figura 49. Logo de MongoDB

Font: <https://www.mongodb.com/brand-resources>

9.1.3. Java

Java [40] és un dels llenguatges de programació més utilitzats per a desenvolupar aplicacions en Android. És un llenguatge de programació orientat a objectes basat en classes.

S'ha decidit utilitzar Java ja que està suportat per Android Studio i, a diferència de Kotlin que també està suportat, ja tenia experiència prèvia desenvolupant aplicacions d'Android amb Java.

Tot el codi del front-end de l'aplicació està fet amb Java.



Figura 49. Logo de Java

Font: <https://www.java.com>

Les llibreries més importants de Java que s'han utilitzat són:

- **Retrofit [41].** Aquesta llibreria permet transformar una API en una interfície de Java, per tal de poder fer-hi crides asíncrones, és a dir, crides que no bloquegen el programa mentre s'executen, de tal manera que l'execució del programa segueix independentment de si la crida ha finalitzat o no.

Com s'ha vist en els diagrames de l'apartat d'arquitectura del sistema, s'utilitza una instància de la classe Retrofit per a crear els diferents serveis i fer totes les crides a l'API que es comunica amb el back-end.

Una vegada finalitza la crida, es tracten els resultats mitjançant un *callback*.

- **Glide [42]**. Glide és una llibreria de gestió d'imatges i vídeos, que s'encarrega de descodificar, guardar en memòria i utilitzar la cache del disc per gestionar les imatges i vídeos d'una aplicació Android. En aquesta aplicació s'ha utilitzat per a totes les imatges que es mostren, ja que redueix el temps de càrrega d'una manera considerable.
- **Google Play Services [43]**. Aquesta llibreria permet accedir als serveis de Google Play d'una manera fàcil i ràpida. En aquest projecte s'utilitza per gestionar l'inici de sessió de l'usuari.

9.1.4. XML

XML (o Extensible Markup Language) [44] és "un llenguatge de marques com HTML dissenyat per emmagatzemar i transportar dades", segons les definicions de *w3schools* [45].

En l'aplicació d'Android s'ha utilitzat per fer les diferents vistes i recursos visuals.

9.1.5. AndroidX

Segons la documentació oficial d'Android, AndroidX és "un projecte de codi obert que l'equip d'Android utilitza per desenvolupar, provar, empaquetar i crear versions de les biblioteques Jetpack" [46], que són "un conjunt de biblioteques, eines i guies per ajudar als desenvolupadors a escriure apps d'alta qualitat de manera més senzilla" [47]

A efectes pràctics, la documentació oficial d'Android diu que "AndroidX reemplaça les API de compatibilitat original amb paquets en l'espai de noms de androidx" [48].

En definitiva, és una nova estructura de paquets que pretén aclarir quins paquets formen part d'Android i quins no.

S'ha decidit migrar el projecte a AndroidX perquè hi havia certes funcions necessàries per al correcte funcionament de l'aplicació que només estaven disponibles en les versions d'AndroidX dels paquets. A més, en el futur, molts paquets que no són d'AndroidX quedaran deprecats, pel que és una migració que eventualment hagués sigut necessària.

9.2. Eines de desenvolupament i altres serveis

9.2.1. PyCharm

PyCharm [49] és l'eina de desenvolupament integrada (IDE) de Python de JetBrains. Tot i que hi ha altres IDEs de Python, s'ha decidit per PyCharm per l'experiència prèvia de la qual dispo amb aquest programa.



Figura 50. Logo de PyCharm

Font: <https://www.jetbrains.com/pycharm/>

PyCharm s'ha utilitzat per fer els dos script del back-end i l'API que comunica el front-end amb el back-end.

9.2.2. Android Studio

Android Studio [50] és l'IDE oficial per a desenvolupar aplicacions d'Android.



Figura 51. Logo de Android Studio

Font: <https://www.android.com/>

Hi havia altres opcions per a desenvolupar aplicacions per a Android, com per exemple Ionic [51], que permet desenvolupar aplicacions per a iOS i Android a la vegada. Tot i això, s'ha decidit per utilitzar Android Studio per l'experiència prèvia de la qual dispo de el programa, mentre que utilitzant Ionic hagués hagut d'aprendre des de zero.

9.2.3. MongoDB Atlas

MongoDB Atlas [20] és un servei de *hosting* de base de dades de MongoDB. Aquest servei permet emmagatzemar, crear, editar, consultar i eliminar dades de la base de dades a més de proporcionar una interfície gràfica per visualitzar aquestes dades.

S'ha utilitzat per allotjar la base de dades del sistema.

9.2.4. Heroku

Segons la mateixa web de Heroku, Heroku és "una plataforma en el núvol que permet a companyies crear, entregar, monitorar i escalar aplicacions" [52]. És a dir, una plataforma que ofereix servidors per allotjar aplicacions.

Heroku s'ha utilitzat per a allotjar l'API que comunica el front-end amb el back-end.

9.3. Aspectes rellevants del codi de la implementació

9.3.1. Extracció de dades

Com s'ha dit al llarg del document, les dades del sistema s'extreuen de l'API de PandaScore.

Les dades que hi ha disponibles són:

- Competicions (amb els equips que hi participen).
- Equips.
- Jugadors.
- Resources (campions i mapes).
- Canvis d'equip per part dels jugadors.
- Pròximes sèries.

S'ha desenvolupat un script en Python que manté actualitzada la base de dades amb la informació obtinguda de l'API de PandaScore. Com s'ha explicat en el capítol 8.1, es planejava que aquest script s'executés automàticament cada 8 hores, però de moment s'executa manualment una vegada al dia degut a les dificultats trobades a l'hora d'aconseguir executar automàticament l'script mitjançant Heroku.

En la figura 52 es pot veure un fragment de codi on s'afegeix un jugador nou a la base de dades, on *p_id* és l'id del jugador i *team* és l'equip del jugador.

```
def add_player(p_id, team):
    player_db = db_players.find_one({"_id": p_id})
    # Add player to DB if does not exist
    if player_db is None:
        player = ps.all_games.get_player(p_id)
        if player['first_name'] is None or player['last_name'] is None:
            real_name = None
        else:
            real_name = (player['first_name'] + " " + player['last_name'])
        db_players.insert_one({
            "_id": player['id'],
            "name": player['name'],
            "real_name": real_name,
            "game": player['current_videogame']['name'],
            "image": player['image_url'],
            "country": player['hometown'],
            "slug": player['name'].lower(),
            "team": SON([("id", team['id']), ("name", team['name']), ("abb", team['acronym'])]),
            "stats": []
        })
```

Figura 52. Afegir jugador a la base de dades

Font: Elaboració pròpia

L'únic problema que ha estat necessari solucionar han estat les inconsistències en els canvis d'equip. Les pàgines dels equips de l'API de PandaScore tenen un camp amb els jugadors dels equips, però aquesta informació està desactualitzada en molts dels casos. Per

tant, els jugadors dels equips s'extreuen directament de les pàgines de les competicions, on hi ha els jugadors actualitzats de cada equip. El problema ve quan hi ha un equip que participa en més d'una competició a la vegada, i és possible que hi hagi certes inconsistències.

Per exemple, hi ha dos equips que participen a dues competicions diferents. En la primera, un jugador forma part del primer equip, però en la segona, forma part del segon equip. Tal com estava plantejat l'script al principi, el jugador formarà part només de l'equip de l'última competició tractada, tot i que és possible que sigui fals.

Per resoldre aquest problema, es guarden tots els canvis d'equip durant el transcurs del programa (equip origen, equip destí) i, al final, es decideix quin és l'equip real del jugador de la següent manera:

1. Es guarden les segones instàncies de canvi d'un mateix jugador a *duplicated*.
2. Per a cada instància de *duplicated*:
 - a. Es troba l'equip actual *current_team* del jugador segons l'API de PandaScore (que en la majoria dels casos no està actualitzat).
 - b. Si *current_team* és igual a l'equip destí:
 - i. Si la instància no té equip origen, el jugador no té equip.
 - ii. Si la instància té equip origen, l'equip destí del jugador és l'equip origen.
 - iii. Si la instància té equip destí, es treu el jugador de l'equip destí.

En resum, aquest algorisme reverteix un canvi d'equip si es donen les condicions necessàries. Òbviament no és perfecte, ja que suposa que la informació de l'API de PandaScore no està actualitzada, però funciona en la gran majoria dels casos.

9.3.2. HTML Scraping

Web Scraping és “la pràctica d'utilitzar un programa d'ordinador per examinar una pàgina web i reunir les dades necessàries en el format més adient mentre que es preserva l'estructura de les dades”, segons Kenneth Reitz i Real Python [53]. Per tant, *HTML Scraping* es basa a extreure dades del HTML d'una pàgina web.

S'ha decidit dedicar un apartat sencer a aquesta tècnica ja que ha sigut una pràctica molt important en aquest projecte, ja que la gran majoria d'estadístiques s'han extret fent servir *HTML Scraping*.

Com s'ha comentat en altres apartats, l'API externa de PandaScore no ens dona tota la informació necessària per tenir un sistema d'una qualitat decent. De fet, sí que la dona, però a un preu de 150 € per joc per mes [54], que seria un total de 300 € per mes en aquest projecte, preu que s'escapa totalment del pressupost del projecte.

Per aquest motiu ha sigut necessari trobar una altra manera d'obtenir aquestes dades, que és el *HTML Scraping*, que s'ha utilitzat per extreure les estadístiques de les partides i sèries. Amb aquesta tècnica només és necessari trobar una manera fàcil de tenir un llistat amb links a les pàgines d'estadístiques de les partides de les diferents competicions que es vulguin cobrir, què és precisament el que ofereixen les pàgines web de *Leaguepedia* [55] i *HLTV* [56] per a LoL i CSGO respectivament.

No s'ha trobat cap manera eficaç i fiable d'automatitzar el procés de cerca d'aquests llistats, pel que simplement s'afegeix l'URL manualment en el document de la competició de la base de dades ("link" en la figura 53).

```
_id: 2526
name: "World Championship - 2019 - Group A"
game: "LoL"
> teams: Array
last: 50
link: "https://lol.gamepedia.com/2019_Season_World_Championship/Main_Event/Ma..."
finished: true
> series: Array
```

Figura 53. Captura de pantalla del document d'una competició

Font: <https://cloud.mongodb.com/>

Després, de manera automàtica, s'itera per aquest llistat de links i es tracten les dades trobades a cada una de les partides (tenint en compte de si és una partida de CS:GO, LoL o LoL de la lliga xinesa, com es veu a la figura 54). A més, es guarda un integer en el document de la competició de la base de dades ("last" en la figura 53) que indica l'última partida de la llista que s'ha tractat per tal de saber quines partides ja s'han afegit a la base de dades i quines no, en cas que la competició encara no hagi acabat. En la figura 55 hi ha un exemple de codi de *HTML Scraping* on es troben els campions d'un dels equips.

```

def call_result(url, t_id):
    if url.find("lol.gamepedia.com") != -1:
        print("LoL match succesfully detected")
        try:
            links = get_links_lol(url)
        except:
            return
        if len(links) > 0:
            if links[0][0].find("matchhistory") != -1:
                # LoL Games
                get_stats_lol(links, "normal", t_id)
            else:
                # LoL Games from China (different website)
                get_stats_lol(links, "lpl", url)
        elif url.find("hltv.org") != -1:
            print("CSGO match succesfully detected")
            links = get_links_csgo(url)
            get_stats_csgo(links, url)
        else:
            print("Error: Not a LoL or CSGO match detected")
    return

```

Figura 54. Distingir entre partides de CS:GO, LoL o LoL de Xina amb la URL
Font: Elaboració pròpia

```

main = dr.find_element_by_xpath('//div[@id="main"]')
web = main.get_attribute('innerHTML')
tree = html.fromstring(web)

champions = tree.xpath('//div[@class="champion-icon binding"]')
champions1 = champions[0:5]
champion_picks = []
for c in champions1:
    m = re.search('data-rg-id="(.+?)">', str(etree.tostring(c, pretty_print=True)))
    champion_picks.append(m.group(1))

```

Figura 55. Exemple de HTML Scrapping
Font: Elaboració pròpia

Per a cada partida, s'afegeixen els resultats a la BD i s'actualitzen les estadístiques dels jugadors i equips que hi han participat.

Òbviament, el fet que s'estiguin extraient dades de dos llocs diferents (API de PandaScore i webs externes utilitzant *HTML Scrapping*) dóna lloc a certes possibles inconsistències i problemes que compliquen molt l'automatització del procés d'obtenció de dades. La majoria de problemes són causats pel fet que no es poden buscar els jugadors per id, sinó que s'han de buscar per nom.

Algunes de les coses que han causat més problemes durant la implementació del sistema són:

- **Canvi de nom del jugador.** De vegades un jugador es canvia el nom i no es veu reflectit a l'API de PandaScore, o, simplement, té un caràcter especial en el nom que no està a l'API. Per exemple, el jugador "Razork" ha jugat una partida amb el nom de "Razørk", pel que no s'ha pogut trobar a la base de dades. Per solucionar aquest problema, s'ha inclòs un camp "Alisases" en els jugadors, on es posen de manera manual altres noms amb els quals ha jugat el jugador, per tal que la pròxima vegada que juga amb aquell nom, el sistema ho reconegui de manera automàtica.
- **Dos jugadors tenen el mateix nom.** La solució és buscar els jugadors per nom i equip.
- **Un dels jugadors no forma part de l'equip en la base de dades.** S'afegeix el jugador a la BD.
- **La imatge d'un jugador està desactualitzada.** Per a cada jugador que es tracta, es comprova si la seva imatge encara està disponible. Si no ho està, es canvia automàticament per la nova.
- **La pàgina d'estadístiques d'una partida de LoL està en un altre idioma.** Si passa això, l'HTML Scraping no funcionarà correctament, ja que molts dels elements tindran un nom diferent, pel que cal canviar l'idioma de la web. En la figura 56 es veu la solució d'aquest problema utilitzant manipulació de Strings per canviar l'URL de la pàgina si es detecta que no està en anglès.
- **Distingir sèries de LoL.** En HLTV, la pàgina per a CSGO, tenim un element per sèrie i les diferents partides dins d'aquest element, pel que és automàtic saber quines partides corresponen a cada sèrie. En canvi, a Leaguepedia no passa el mateix. Hi ha un llistat amb totes les partides sense distingir les sèries. Per resoldre aquest problema, es suposa que totes les partides consecutives en les quals juguen els mateixos equips formen part de la mateixa sèrie. En la figura 57 en veiem un exemple, on les 5 primeres partides corresponen a la mateixa sèrie mentre que les dues últimes són dues sèries diferents d'una partida cadascuna.
- **Partides de la lliga xinesa.** La lliga xinesa de LoL utilitza una pàgina web completament diferent de la resta per les seves sèries, on cada pàgina mostra la primera partida d'una sèrie i cal clicar dins de la pàgina per arribar a les altres partides de la sèrie. Per sort, el navegador de Selenium deixa clicar enllaços, però ha estat necessari fer un petit algorisme que descobreix quantes partides hi ha en cada sèrie.
- **Noms dels jugadors de la lliga xinesa.** En totes les lligues els noms dels jugadors segueixen el format "[sigles de l'equip] [nom del jugador]", mentre que en la lliga xinesa segueixen el format "[sigles de l'equip][nom del jugador]" sense espai. Per saber el nom d'aquests jugadors s'ha agafat els cinc strings de tots els jugadors i s'han analitzat quins caràcters es repeteixen a l'inici, què són les sigles dels equips. Per tant la resta del string és el nom del jugador.

```

if league == "normal":
    serie[0] = serie[0] + '!'
    m = re.search(
        'ht(.+?)://matchhistory.(.+?).leagueoflegends.com/(.+?)#match-details/ESPORTSTMNT(.+?)/(.+?)\?gameHash=(.+?)!',
        serie[0])
    if m.group(2) != "na":
        serie[0] = 'https://matchhistory.na.leagueoflegends.com/en/#match-details/ESPORTSTMNT' + m.group(
            4) + '/' + m.group(5) + '?gameHash=' + m.group(6)
    else:
        serie[0] = serie[0][:-1]

```

Figura 56. Modificar URL de la pàgina
Font: Elaboració pròpia















2019-12-26	9.23		
2019-12-26	9.23		
2019-12-26	9.23		
2019-12-26	9.23		
2019-12-26	9.23		
2019-12-22	9.23		
2019-12-22	9.23		

Figura 57. Exemple de sèries a Leaguepedia
Font: https://lol.gamepedia.com/Demacia_Cup_2019/Match_History

9.3.3. API

Per comunicar el front-end amb el back-end s'ha desenvolupat una REST API amb Flask. En la figura 58 es pot veure un exemple del codi d'una de les crides.

```

@app.route('/comments/<comment_id>/replies', methods=['GET'])
def get_replies(comment_id):
    order = request.args['order']
    replies = list(mongo.db.comments.find({"parent_id": comment_id}))
    if order == "points":
        sorted_replies = sorted(replies, key=itemgetter(order), reverse=True)
    else:
        sorted_replies = sorted(replies, key=itemgetter('date', 'time'), reverse=True)
    return Response(json.dumps(sorted_replies), mimetype='application/json')

```

Figura 58. Crida de l'API per aconseguir les respostes d'un comentari
Font: Elaboració pròpia

9.3.4. Fragments

Pel que fa al front-end, s'han utilitzat fragments en diverses vistes. Els fragments són útils per separar les funcionalitats d'un tros de la vista de l'activitat de la vista, com poden ser el cas de pestanyes, tal com s'ha vist en el capítol 8 de la memòria i s'exemplifica a la figura 59.

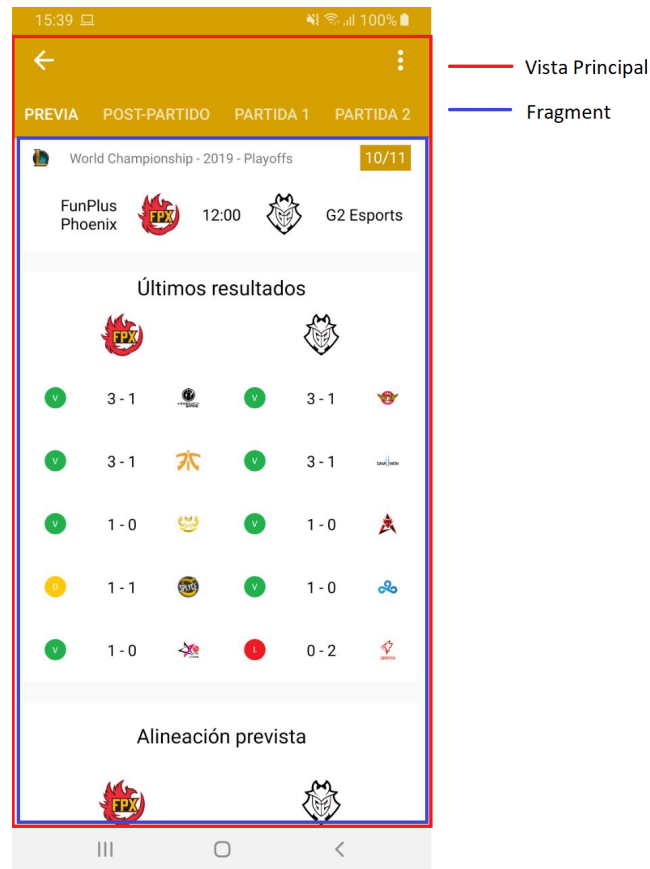


Figura 59. Exemple d'un fragment dins d'una vista
Font: Elaboració pròpia

Els fragments d'una vista es creen des de l'activitat de la vista però, en aquest projecte, els fragments pertanyents a pestanyes es creen des de un adaptador separat de la vista, per tal de simplificar el codi. En la figura 60 es pot veure un fragment del codi de l'adaptador pertanyent a la vista de la figura 59, en el que es retorna un fragment diferent depenent de la pestanya seleccionada.

```

@Override
public Fragment getItem(int position) {
    String type = tabs.get(position);
    switch(type){
        case "pre":
            fragment = new PrematchFragment();
            break;
        case "post":
            fragment = new PostmatchFragment();
            break;
        case "game":
            if(game.equals("LoL"))
                fragment = new GameLolFragment();
            else if(game.equals("CS:GO"))
                fragment = new GameCsgoFragment();
            Bundle args = new Bundle();
            args.putInt("game_number", position-1);
            fragment.setArguments(args);
            break;
    }
    return fragment;
}

```

*Figura 60. Exemple d'adaptador de fragment
Font: Elaboració pròpia*

9.3.5. RecyclerView

Un dels principals mals de cap d'aquest projecte ha estat com mostrar una llista d'elements, quan no se sap el nombre d'elements. Per sort, en Android existeixen les RecyclerView, llistes d'elements indefinits.

Perquè funcioni correctament és necessari crear un adaptador per a la llista d'elements que es vol mostrar. Aquest adaptador necessita una subclasse que representa cadascun dels elements de la llista (figura 61) i després popular els elements des de la classe principal de l'adaptador (figura 62). Els exemples mostrats en les figures 61 i 62 representen un adaptador per mostrar les sèries de la vista *UpcomingMatches* que s'ha vist anteriorment en la figura 13, és a dir, aquelles sèries que encara no han començat i l'atribut series representa la llista de les sèries que es vol mostrar.

Cal dir que els comentaris és un cas una mica especial, ja que cada comentari té el seu propi RecyclerView per mostrar les rèpliques al comentari.

```

public class MyViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {
    TextView tournament_name, date, time, team1_name, team2_name;
    Serie s;
    ImageView game_icon, team1_icon, team2_icon;

    MyViewHolder(View view) {
        super(view);
        tournament_name = view.findViewById(R.id.tournament_name);
        date = view.findViewById(R.id.date);
        time = view.findViewById(R.id.hour);
        team1_name = view.findViewById(R.id.team1_name);
        team2_name = view.findViewById(R.id.team2_name);
        game_icon = view.findViewById(R.id.icon_game);
        team1_icon = view.findViewById(R.id.icon_team1);
        team2_icon = view.findViewById(R.id.icon_team2);
        view.setClickable(true);
        view.setOnClickListener(this);
    }
}

```

*Figura 61. ViewHolder d'una sèrie
Font: Elaboració pròpia*

```

@Override
public void onBindViewHolder(MyViewHolder holder, int position) {
    Serie s = series.get(position);
    holder.date.setText(s.getDateReduced());
    holder.time.setText(s.getTimeReduced());
    holder.tournament_name.setText(s.getTournament_name());
    holder.team1_name.setText(s.getTeam1_name());
    holder.team2_name.setText(s.getTeam2_name());
    ImageView team1_icon = holder.team1_icon;
    Glide.with(context).load(series.get(position).getTeam1_icon()).into(team1_icon);
    ImageView team2_icon = holder.team2_icon;
    Glide.with(context).load(series.get(position).getTeam2_icon()).into(team2_icon);
    ImageView game_icon = holder.game_icon;
    Glide.with(context).load(series.get(position).getGame_icon()).into(game_icon);
    holder.s = s;
}

```

*Figura 62. Popular un ViewHolder
Font: Elaboració pròpia*

9.3.6. Crides a l'API

Per cridar a l'API s'ha utilitzat la llibreria externa de Retrofit, que permet establir connexions HTTP. En la figura 63 es mostra un exemple d'una de les interfícies fetes per aquestes crides. S'ha creat una interfície per a cada concepte del sistema, per poder tenir les crides ben organitzades.

Aquestes crides es criden des dels diferents controladors de domini, tal com es mostra en la figura 64. També s'hi tracten els resultats.

```
public interface TeamService {
    @GET("/teams/{team_id}/last_results")
    Call<List<JsonElement>> getLastResults(@Path(value = "team_id", encoded = true) String teamId, @Query(value = "date") String date);

    @GET("/teams/{team_id}/lineup")
    Call<List<JsonElement>> getLineup(@Path(value = "team_id", encoded = true) String teamId, @Query(value = "date") String date);

    @GET("/teams/{team_id}")
    Call<Team> getTeam(@Path(value = "team_id", encoded = true) String teamId);
}
```

*Figura 63. Interfície del servei d'equips
Font: Elaboració pròpia*

```
void getTeam(String team_id) {
    TeamService service = RetrofitClientInstance.getRetrofitInstance().create(TeamService.class);
    Call<Team> call = service.getTeam(team_id);
    call.enqueue(new Callback<Team>() {
        @Override
        public void onResponse(Call<Team> call, Response<Team> response) {
            cp.showTeam(response.body());
        }

        @Override
        public void onFailure(Call<Team> call, Throwable t) { Log.e( tag: "retrofit", t.toString()); }
    });
}
```

*Figura 64. Exemple de crida a l'API per trobar un equip determinat
Font: Elaboració pròpia*

9.3.7. Google Sign In

S'ha utilitzat el servei de Sign In de Google per iniciar sessió al sistema. Aquest servei proporciona el compte de Google de l'usuari.

Aquest compte s'envia al controlador de domini determinat i, com es veu en la figura 65, primer es comprova si aquest usuari ja existeix a la base de dades del sistema o, si pel contrari, és la primera vegada que inicia sessió. En el primer cas, simplement es crea la sessió en el sistema per a aquest usuari, mentre que en el segon, primer es crea un usuari en el sistema amb les dades del seu compte de Google i s'afegeix a la base de dades mitjançant una crida a l'API (figura 66).

L'únic problema que ha calgut resoldre aquí és en el cas que l'usuari no tingui imatge de perfil a Google. Llavors, simplement es posa una imatge sense copyright que representa que l'usuari no té imatge de perfil.

```
void getUser(GoogleSignInAccount account) {
    String id = account.getId();
    UserService service = RetrofitClientInstance.getRetrofitInstance().create(UserService.class);
    Call<User> call = service.getUser(id);
    call.enqueue(new Callback<User>() {
        @Override
        public void onResponse(Call<User> call, Response<User> response) {
            if(response.body() == null) createUser(account);
            else{
                createSession(response.body());
            }
        }

        @Override
        public void onFailure(Call<User> call, Throwable t) { Log.e("retrofit", t.toString()); }
    });
}
```

Figura 65. Comprovar si l'usuari existeix

Font: Elaboració pròpia

```
private void createUser(GoogleSignInAccount account) {
    Uri image_url = account.getPhotoUrl();
    String image;
    if(image_url == null) image = "https://genslerzudansdentistry.com/wp-content/uploads/2015/11/anonymous-user.png";
    else image = image_url.toString();
    User user = new User(account.getId(), account.getDisplayName(), image, type: 0,
        Locale.getDefault().getLanguage(), new ArrayList<>(), new ArrayList<>(), spoilers: false);
    UserService service = RetrofitClientInstance.getRetrofitInstance().create(UserService.class);
    Call<Void> call = service.createUser(user);
    call.enqueue(new Callback<Void>() {
        @Override
        public void onResponse(Call<Void> call, Response<Void> response) {
            createSession(user);
        }

        @Override
        public void onFailure(Call<Void> call, Throwable t) {
        }
    });
}
```

Figura 66. Crear nou usuari

Font: Elaboració pròpia

10. Proves i validació

Tots els sistemes necessiten un seguit de proves i una fase de validació per comprovar que funciona com s'esperava i assoleix un mínim de qualitat. En aquest capítol s'expliquen els mètodes de validació seguits i les proves fetes.

Per començar, s'han seguit els mètodes de validació presentats a la secció 4.3:

- Validació de tasques

S'han implementat una sèrie de tests automàtics unitaris i d'integració per a les tasques més importants del projecte, i s'ha retocat el codi pertanyent a les tasques fins que aquests tests passaven correctament. En altres paraules, s'han fet els tests independentment del codi de l'aplicació, per tal de garantir que el codi funciona com s'esperava.

Aquests tests automàtics s'han fet en Android Studio amb la versió 4.12 de la llibreria de Java *JUnit* [60]. Aquesta llibreria permet fer tests automàtics d'una manera senzilla. En la figura 67 es mostra un exemple d'un test unitari, o sigui, un test en el qual es valida que una unitat de software funciona correctament. En aquest exemple en concret es valida que la funció que comprova que un nom d'usuari sigui vàlid funcioni correctament (ha de retornar 0 si és vàlid i un integer entre 1 i 4 si no ho és, depenent de quina regla viola).

```
@Test
public void check_name_rules() {
    cpProfile cp = new cpProfile();
    String username = "test";
    assertEquals( expected: 1, cp.check_name(username, new_name: "test"));
    assertEquals( expected: 2, cp.check_name(username, new_name: ""));
    assertEquals( expected: 3, cp.check_name(username, new_name: "j1_"));
    assertEquals( expected: 3, cp.check_name(username, new_name: "j1*"));
    assertEquals( expected: 4, cp.check_name(username, new_name: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaa"));
    assertEquals( expected: 0, cp.check_name(username, new_name: "correct name"));
}
```

Figura 67. Test unitari

Font: Elaboració pròpia

Per altra banda també s'han fet tests d'integració amb la llibreria *JUnit*. En un test d'integració es comprova que dues unitats de software funcionen correctament juntes. En aquest cas, es comprova que l'aplicació funciona correctament amb l'API que es comunica al back-end. Per solucionar que *Retrofit* funciona amb crides asíncrones i, per tant, no es poden comprovar els resultats de les crides sense esperar als resultats, ha estat necessari una llibreria pròpia de Java anomenada *CompletableFuture* [61]. Aquesta llibreria permet esperar que les crides asíncrones acabin abans de continuar l'execució del codi. Es mostra un exemple d'un test d'integració utilitzant aquesta llibreria en el que es comprova que un jugador es carrega correctament des de la base de dades en la figura 68.


```

@Test
public void loadPlayer() {
    PlayerService service = RetrofitClientInstance.getRetrofitInstance().create(PlayerService.class);
    CompletableFuture<Player> future = new CompletableFuture<>();
    Call<Player> call = service.getPlayer( player_id: "2146");
    call.enqueue(new Callback<Player>() {
        @Override
        public void onResponse(Call<Player> call, Response<Player> response) {
            future.complete(response.body());
        }

        @Override
        public void onFailure(Call<Player> call, Throwable t) {
            Log.e( tag: "retrofit", t.toString());
        }
    });
    assertNotNull(future.join());
    assertEquals( expected: "Rekkles", future.join().getName());
    assertEquals( expected: "Martin Larsson", future.join().getReal_name());
    assertEquals( expected: "LoL", future.join().getGame());
}

```

Figura 68. Test d'integració

Font: Elaboració pròpia

- Validació de sprint

Al final de cada *sprint* s'ha fet una reunió amb la directora del projecte per rebre feedback del treball fet fins al moment. En certes ocasions, s'ha modificat la planificació del següent *sprint* o l'ordre en el qual es fan les tasques per adaptar-se a les limitacions de temps.

- Validació final

Una vegada acabats tots els requisits funcionals del projecte, s'ha comprovat manualment que es compleixen tots els escenaris principals d'èxit i les extensions presentats a l'apartat "Descripció dels casos d'ús".

Per a cada cas d'ús que no complís els escenaris i/o les extensions, s'ha decidit si era més adient modificar el codi o la descripció d'ús, en el cas que no reflectís correctament el que es volia del cas d'ús en qüestió.

- Test d'usabilitat amb altres usuaris

En diferents punts del desenvolupament del sistema s'han fet tests d'usabilitat amb altres usuaris amb diferents nivells de coneixement d'aplicacions de dispositius mòbils, d'e-sports i de competicions esportives en general, per tal de poder millorar sobretot la claredat de la interfície.

Aquests tests han consistit a demanar a un total de 20 usuaris, amb els i coneixements previs i interessos que es mostren a les taules 8 i 9, on en cada cel·la hi apareix el nombre d'usuaris pertinents a aquell grup, que completessin aquestes tasques en aquest ordre dins de l'aplicació:

- Obre l'última sèrie jugada de CSGO.
- Busca informació sobre qualsevol jugador de *LoL*.
- Inicia sessió
- Marca qualsevol equip com a preferit.
- Filtra les sèries per sèries dels teus equips preferits.
- Activa el mode sense spoilers.
- Canvia l'idioma de l'aplicació.
- Tanca sessió.

Pocs coneixements d'aplicacions de dispositius mòbil	Coneixements mitjans o alts d'aplicacions de dispositius mòbil
2	18

Taula 8. Coneixements previs dels usuaris enquestats.

Font. Elaboració pròpia

	Interessat/da en els e-sports	No interessat/da en els e-sports
Utilitza aplicacions de resultats esportius	3	4
No utilitza aplicacions de resultats esportius	9	4

Taula 9. Interessos dels usuaris enquestats.

Font. Elaboració pròpia

Si l'usuari és incapaç de dur a terme alguna tasca d'una manera ràpida relativa als seus coneixements informàtics previs, vol dir que la interfície està mal plantejada i s'ha de repensar.

Per sort, els únics usuaris que han tingut problemes amb aquestes tasques eren els usuaris amb un coneixement previ baix o nul d'aplicacions mòbil, i tots han estat capaços de dur a terme les tasques després d'una breu explicació de com funciona una interfície mòbil (per exemple, que la icona amb tres línies horitzontals sol representar un menú lateral en les aplicacions mòbil).

Cap altre usuari ha tingut problemes en dur a terme les tasques proposades, tot i que els usuaris amb un interès previ amb competicions esportives han fet les tasques considerablement més de pressa que la resta d'usuaris. Això es deu a que molts d'aquests usuaris utilitzen aplicacions similars per a altres esports, com poden ser el futbol o el basket.

Pel que fa a l'estil de l'aplicació, és a dir, colors, font, tamany de la lletra, etc. cap usuari va tenir problemes en la correcta visualització del contingut més enllà de preferències personals del color. Alguns dels usuaris van proposar el

desenvolupament d'un tema fosc per l'aplicació. Aquesta proposta es va considerar, però finalment es va descartar per les restriccions temporals del projecte.

També es va demanar per l'opinió d'un psicòleg, qui no va trobar problemes amb la paleta de colors escollida.

A més d'aquests mètodes de validació, s'han utilitzat els usuaris dels tests d'usabilitat per a validar els requisits no funcionals.

A continuació es mostra, per a cada requisit no funcional presentat a l'apartat de "Requisits no funcional", la classificació del requisit segons Volere, una descripció del requisit, el seu criteri d'acceptació i el resultat i algunes observacions si és necessari de l'estudi fet.

11a. Ease of use Requirement. El sistema ha de ser fàcil d'utilitzar per a qualsevol persona amb un coneixement mínim d'aplicacions mòbil.

Criteri d'acceptació: El 80% dels usuaris han de ser capaços de completar les tasques bàsiques de l'aplicació sense informació prèvia.

Resultats i observacions: Criteri acceptat. Tots els usuaris amb coneixements previs del funcionament d'aplicacions mòbil han estat capaços de completar les tasques bàsiques de l'aplicació. La resta han estat capaços després d'una breu explicació del funcionament bàsic d'una interfície d'una aplicació mòbil

11b. Personalization and Internationalization Requirement. El sistema ha d'estar disponible en diverses llengües.

Criteri d'acceptació: L'aplicació estarà disponible en castellà i anglès.

Resultats i observacions: Criteri acceptat.

11b. Personalization and Internationalization Requirement. El sistema ha de recordar la configuració bàsica de l'usuari.

Criteri d'acceptació: L'aplicació guarda la configuració bàsica de l'usuari (idioma i mode spoilers) en el 100% dels casos.

Resultats i observacions: Criteri acceptat.

11b. Personalization and Internationalization Requirement. Les dates i hores han d'estar en la zona horària de l'usuari

Criteri d'acceptació: L'aplicació mostra les hores en la zona horària de l'usuari. Si l'usuari canvia la zona horària del dispositiu, l'aplicació reflecteix els canvis.

Resultats i observacions: Criteri acceptat.

11d. Understandability and Politeness Requirement. Els termes utilitzats per el sistema han de ser comprensibles per la comunitat d'usuaris a la que orientat el sistema.

Criteri d'acceptació: El 80% dels usuaris que formen part de la comunitat d'usuaris a la que va destinada l'aplicació (és a dir, els usuaris que han jugat als jocs que ofereix), entenen els termes específics que s'utilitzen.

Resultats i observacions: Criteri acceptat. En l'estudi fet, tots els usuaris amb coneixements previs als jocs han entès els termes específics utilitzats.

11e. Accesibility Requirement. El sistema ha de ser usable persones amb diferents discapacitats visuals, com pot ser el daltonisme.

Criteri d'acceptació: Diverses persones amb discapacitats visuals poden utilitzar el sistema sense dificultats.

Resultats i observacions: Criteri acceptat. S'ha preguntat a dues persones amb daltonisme.

12a. Speed and Latency Requirement. El sistema no trigarà més de 5 segons en mostrar cap pantalla amb una connexió superior als 10Mb/s.

Criteri d'acceptació: Es farà un test del que triga en carregar les diverses pantalles amb diverses connexions a Internet.

Resultats i observacions: Criteri acceptat. S'ha provat amb diferents velocitats de connexió (30 Mb/s, 98 Mb/s i 612 Mb/s) i el temps màxim d'espera ha estat de 2,5 segons. Cal dir que la primera connexió després d'una hora de no ús és d'un temps més elevat (15 segons), ja que la versió gratuïta del servidor de Heroku entre en mode dormit. Això es pot arreglar comprant una versió premium del servidor.

12d. Robustness or Fault-Tolerance Requirement. En cas d'error, l'aplicació no haurà de penjar-se.

Criteri d'acceptació: L'aplicació no es penja en casos extrems.

Resultats i observacions: Criteri acceptat.

13b. Requirement for Interfacing with Adjacent Systems. L'aplicació ha de funcionar en la gran majoria de dispositius Android.

Criteri d'acceptació: L'aplicació ha de funcionar com a mínim en el 80% dels dispositius Android.

Resultats i observacions: Criteri acceptat. En tots els dispositius Android que s'ha provat ha funcionat.

11. Aspectes legals

Aquest capítol es centra en els aspectes legals del sistema. En la secció 11.1 es parla sobre la legalitat d'utilitzar els diferents recursos de *League of Legends* i *Counter Strike: Global Offensive*. Aquests recursos inclouen logos dels jocs, imatges dels seus campions i mapes i logos i noms dels equips i jugadors. A més, és important complir amb totes les llicències que es requereixen per al projecte, tal com s'explica en la secció 11.2.

11.1. Lleis aplicables

Comencem per *League of Legends*. En els termes legals de la pàgina de Riot Games [62] es diu que els únics projectes vàlids són projectes no comercials, és a dir, que no generen diners, amb l'excepció de la generació d'ingressos passiva de la publicitat. En principi, no hi hauria problema amb aquesta part, però més endavant s'especifica que no s'autoritzen jocs o aplicacions en la Apple Store o en la Google Play Store a no ser que es tingui consentiment escrit o una clau de l'API de Riot vàlida i que el projecte compleixi amb els termes [63] i polítiques [64] de l'API de Riot.

Aquest projecte no utilitza l'API de Riot pel que s'entén que es necessita un escrit legal per part de Riot, pel que no es publicarà l'aplicació a la Google Play Store fins que no s'aconsegueixi, però no hi hauria d'haver problemes quan es compara l'aplicació amb aquelles vistes a l'estat de l'art i veient els requisits bàsics dels termes legals (no ha de ser un projecte comercial i ha de ser original, de la mateixa pàgina web: "Coses interessants i gratuïtes perquè la comunitat gaudeixi, amb algunes excepcions").

Pel que fa a *Counter Strike*, no s'ha trobat cap pàgina web similar a l'anterior on es digui clarament que es permet i que no es permet amb la seva IP. El més semblant és la pàgina d'informació legal de *Steam* [65], en la que es diu que *Counter Strike* i el seu logo són marques registrades. Com abans, tenint en compte les aplicacions i webs trobades a l'estat de l'art, aquest projecte no hauria de tenir cap problema.

Per resumir, es considera que aquesta aplicació cau sota l'ús raonable (o *fair use*) tal com es descriu a la secció 107 de la Llei del Dret d'Autor dels Estats Units [66], en la que es diu si es pot utilitzar una obra que té drets d'autor considerant els següents factors:

- El propòsit i caràcter de l'ús, incloent-hi si aquest ús és de naturalesa comercial o per propòsits instructius sense ànim de lucre;
- La classe del dret d'autor de l'obra;
- La quantitat i consistència de la porció utilitzada en relació amb el dret d'autor de l'obra;
- L'efecte del seu ús sobre el mercat potencial o valor del dret de l'autor de l'obra.

Aquesta és una aplicació no comercial, on només es generaran ingressos mitjançant la generació d'ingressos passiva de la publicitat.

La finalitat de l'aplicació és donar informació sobre les diferents competicions dels jocs d'una manera ràpida i senzilla, pel que és necessari tenir certs aspectes de l'obra del joc, com poden ser els logos o les imatges dels campions i mapes, que és una quantitat molt petita sobre l'obra total.

A més, no es considera que aquesta aplicació tingui un efecte negatiu sobre l'obra original dels jocs, ja que no pretén reemplaçar-la sinó complementar-la.

El mateix s'aplica als logos i imatges dels jugadors i equips de les diferents competicions.

Tot i això, no es publicarà l'aplicació a la Google Play Store fins que no s'aconsegueixi un consentiment per part de les empreses dels jocs.


Pel que fa a les dades guardades dels usuaris, cal complir amb l'anomenat RGPD o Reglament General de Protecció de Dades de la Unió Europea [67]. Aquest sistema no guarda cap mena de contrasenya dels usuaris, ja que utilitza els serveis de Google Play per iniciar sessió. A més, les dades dels usuaris es guarden en *Atlas MongoDB*, que garanteix a la seva pàgina web [68] que encripta les dades utilitzant TLS/SSL, un protocol per mantenir una connexió a Internet segura [69].

11.2. Llicències

Abans de mostrar les llicències utilitzades en aquest projecte, convé definir les llicències permissives més bàsiques que s'esmentaran i es referenciaran:

- **Llicència MIT.** Una llicència MIT és una llicència de software permissiva i gratuïta. Permet utilitzar el software per a projectes comercials i es pot modificar i distribuir [70]. En la figura 69 es mostra un exemple d'aquesta llicència.
- **Llicència Apache (2.0).** Aquesta llicència també és una llicència permissiva i gratuïta. Permet utilitzar el software, distribuir-ho i modificar-ho [71].
- **Llicències BSD.** Les llicències BSD són llicències de software permissives i gratuïtes. Permet redistribució i ús del software si les redistribucions tenen l'avís de *copyright* i no s'utilitza el nom de l'autor per promoure productes derivats sense permís escrit. [72]
- **GPL.** La llicència GPL indica que el software es pot utilitzar per a qualsevol propòsit, es pot modificar per adaptar-lo a les seves necessitats, es pot compartir el programa amb els seus amics i veïns i es poden compartir les millores que hi aporti. [73]

Totes aquestes llicències permeten utilitzar el software proporcionat sense (o amb molt poques) restriccions, pel que s'han intentat buscar serveis de software que les utilitzin durant la planificació i el desenvolupament del projecte.

 <p>LucasCLuk/pandascore is licensed under the MIT License</p> <p>A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.</p>	<p>Permissions</p> <ul style="list-style-type: none"> ✓ Commercial use ✓ Modification ✓ Distribution ✓ Private use 	<p>Limitations</p> <ul style="list-style-type: none"> ✗ Liability ✗ Warranty 	<p>Conditions</p> <ul style="list-style-type: none"> ① License and copyright notice
---	---	---	---

This is not legal advice. [Learn more about repository licenses.](#)

Figura 69. Llicència MIT de PandaScore

Font: <https://github.com/LucasCLuk/pandascore/blob/master/LICENSE.md>

Tot seguit es defineixen les llicències i els termes i condicions que es considerin rellevants dels productes i serveis utilitzats durant aquest projecte:

- **PandaScore.** Llicència MIT. En els termes d'ús [74] s'especifica que només es poden reproduir les dades de la base de dades de PandaScore per a "fins informatius", que és precisament el que pretén aquest sistema. També esmenta que no es permet "proveir accés directe des del lloc del client a la base de dades, la interfície de PandaScore, o a qualsevol lloc de pandascore.co [...]. El client ha de reproduir [a la seva pàgina web o aplicació] les dades a les quals té accés i que vol disseminar". L'aplicació del projecte garanteix aquesta condició, ja que en cap moment redirigeix a la base de dades de PandaScore o a pandascore.co i reproduceix directament les dades.
- **IntelliJ Idea.** Llicència Apache 2.0. IntelliJ Idea inclou PyCharm i Android Studio.
- **Python.** Llicència Python Software Foundation. Aquesta llicència és una llicència BSD compatible amb la GPL. [75]. Aquesta llicència permet distribuir projectes software fets amb Python (el back-end d'aquest sistema).
- **MongoDB.** Llicència Server Side Public License (SSPL) [76], que està basada en la GPL. La principal diferència és que la SSPL aclareix la condició d'oferir MongoDB com a servei. [77]
- **Selenium.** Llicència Apache 2.0.
- **Flask.** Llicència BSD de tres clauses. Aquesta és la llicència BSD explicada anteriorment i l'original. Alguns serveis fan l'aclariment de les tres clauses perquè existeix una llicència BSD simplificada amb només dues clauses.
- **Retrofit.** Llicència Apache 2.0.
- **Glide.** Llicència BSD.
- **Taiga.** Llicència AGPL 3.0. Aquesta llicència està basada en la GPL i està dissenyada per no disparar les provisions del *copyleft* quan s'utilitza però no es distribueix el software. [78]

12. Seguiment del projecte

En aquest capítol s'expliquen i es justifiquen els canvis de planificació i pressupost del projecte (secció 12.1) i els canvis de les històries d'usuari (secció 12.2).

12.1. Canvis de planificació i pressupost

Tot i que en general el projecte ha seguit la planificació temporal i s'han assolit els objectius que es van marcar a l'inici d'aquest, és veritat que han ocorregut certs problemes que han causat desviacions en la planificació temporal i el pressupost.

Les limitacions de l'API de PandaScore han estat el principal problema. Es va explicar en l'apartat de "Gestió temporal" com un risc amb una alta probabilitat que ocorregués, pel que es va afegir al diagrama de Gantt mostrat en la figura 4. Així i tot, es va subestimar bastant el temps necessari per a fer aquesta part del back-end, ja que es van triar diverses maneres per a fer-ho, fins que es va estimar que l'HTML Scraping era la millor. Una vegada escollida la tècnica, es va modificar el codi diverses vegades al llarg del projecte per millorar-lo o arreglar els problemes propis d'utilitzar la tècnica per primera vegada.

Aquesta és la raó principal per la qual no s'han fet la tasca dels serveis web per al back-end que es van especificar, per falta de temps. El fet d'utilitzar HTML Scraping també ha dificultat que es puguin extreure les dades automàticament cada cert període de temps, ja que requereix tenir instal·lat un programa extern i, per tant, no es pot fer des d'un servidor extern com pot ser Heroku. Ara mateix les actualitzacions de la base de dades es fan a través dels scripts de Python i tots els canvis que vénen provocats per errors o inconsistències en els scripts es fan manualment, ja sigui directament a la base de dades o mitjançant petits scripts de Python.

A més, també es va subestimar el temps de fer les crides a la REST API i en certa manera el temps de desenvolupament del front-end de l'aplicació Android. El primer es deu a tots els problemes pertinents a aconseguir establir una connexió amb l'API, fins que es va trobar la llibreria Retrofit explicada en altres apartats. Les 25 hores preestablertes hagueren estat precises si s'hi hagués sabut de l'existència d'aquesta llibreria des del principi. Pel que fa al front-end, simplement es va subestimar la complexitat de certes tasques.

Les altres tasques si què han seguit més o menys la planificació temporal establerta.

Tot seguit, a la taula 8, es mostren les hores planificades i les reals per a cada tasca mostrada en la taula 1 d'aquesta memòria, a més de la diferència entre les hores planificades i les reals.

Codi	Nom Descriptiu	Temps Estimats (hores)	Temps Real (hores)	Diferència
T1	Definició d'objectius i requisits	45	49	4
T2	Tria i estudi de l'API	25	16	-9
T3	Disseny de la BD	35	30	-5
T4	Integració API a BD	50	60	10
T5	Afegir altra informació a BD	50	95	45
T6	Disseny de l'aplicació Android	35	31	-4
T7	Desenvolupament de REST API	25	26	1
T8	Desenvolupament del front-end de l'aplicació Android	50	65	15
T9	Fer les crides a la REST API	25	47	22
T10	Testing de l'aplicació	45	50	5
T11	Desenvolupament de serveis webs per al back-end	40	0	-40
T12	Gestió del projecte	120	130	10
TOTAL	-	545	599	54

Taula 8. Hores estimades i reals del projecte

Font: Elaboració pròpia

S'ha treballat un total de 599 hores, 54 més de les 545 estimades, el que significa un desviament del 9,91% sobre la planificació temporal. Clarament la poca experiència en fer projectes d'aquesta envergadura i en l'estimació de tasques, ha portat a subestimar moltes de les tasques proposades i a no tenir en compte els problemes que sorgeixen durant la fase de desenvolupament del projecte; moltes de les estimacions eren molt optimistes, quan es veuen després d'acabar el projecte.

Com que existeix una desviació en la planificació del temps, també cal veure les desviacions del pressupost. En el control de gestió es van presentar els diferents mètodes per a veure els desviaments dels recursos humans i les tasques.

Tasca	Hores per rol						Cost (€)
	Cap de projecte	Analista	Arquitecte	Front-end	Back-end	Tester	
Definició d'objectius i requisits	12	37					954,00
Tria i estudi de l'API	3	13					306,00
Disseny de la BD	5		25				570,00
Disseny de l'aplicació Android	5		26				588,00
Integració API a BD					60		960,00
Afegir altra informació a BD					95		1520,00
Desenvolupament de la REST API					26		416,00
Desenvolupament del front-end de l'aplicació Android				65			1040,00
Fer les crides a la REST API				47			752,00
Testing de l'aplicació						50	750,00
Desenvolupament de serveis webs per al back-end							0,00
Reunions amb la directora	40						960,00
Document de definició de l'abast i contextualització	5						120,00
Document de la planificació temporal	8						192,00
Document de la gestió econòmica i sostenibilitat	12						288,00
Document final i preparació de la defensa	65						1560,00
TOTAL	155	50	51	112	181	50	10976,00

Taula 9. Cost real per tasca del projecte
Font: Elaboració pròpia

En la taula 9 podem veure les dades reals que ha dedicat cada rol a cada tasca i el cost que comporta cada tasca. En la taula 3 veiem aquesta mateixa taula amb les hores i els costos estimats, i ara es veu que hi ha certes diferències. A part de l'augment del cost d'algunes tasques a causa de les raons explicades anteriorment, destaca la inclusió de la tasca "Afegir altra informació a la BD". Tot i que aquesta tasca estava inclosa en la planificació temporal, no deixava de ser un risc, pel que en el pressupost es va determinar que requeria 640,00 €. Com que finalment aquest risc s'ha complert, s'han determinat les hores que ha treballat cada rol i s'ha inclòs en el pressupost com a tasca, fet que fa que augmenti el pressupost total del projecte. A partir d'ara, les últimes 5 tasques de la taula 9 correspondran a la tasca general "Gestió del projecte", per tal de facilitar la lectura del document.

En les taules 10 i 11 es mostren els desviaments del cost de les tasques i els diferents rols respectivament.

Nom Tasca	Cost Estimat (€)	Cost Real (€)	Desviament (€)
Definició d'objectius i requisits	870,00	954,00	84,00
Tria i estudi de l'API	480,00	306,00	-174,00
Disseny de la BD	660,00	570,00	-90,00
Integració API a BD	800,00	960,00	160,00
Afegir altra informació a BD	640,00	1520,00	880,00
Disseny de l'aplicació Android	660,00	588,00	-72,00
Desenvolupament de REST API	400,00	416,00	16,00
Desenvolupament del front-end de l'aplicació Android	800,00	1040,00	240,00
Fer les crides a la REST API	400,00	752,00	352,00
Testing de l'aplicació	675,00	750,00	75,00
Desenvolupament de serveis webs per al back-end	640,00	0,00	-640,00
Gestió del projecte	2880,00	3120,00	240,00
TOTAL	9905,00	10976,00	1071,00

Taula 10. Desviacions del cost de les tasques

Font: Elaboració pròpia

Rol	Hores Estimades	Hores Reals	Cost per hora (€)	Cost Estimat (€)	Cost Real (€)	Desviament (€)
Cap de projecte	145	155	24,00	3480,00	3720,00	240,00
Analista	55	50	18,00	990,00	900,00	-90,00
Arquitecte	60	51	18,00	1080,00	918,00	-162,00
Front-end	75	112	16,00	1200,00	1792,00	592,00
Back-end	115	181	16,00	1840,00	2896,00	1056,00
Tester	45	50	15,00	675,00	750,00	75,00
TOTAL	495	599	107,00	9265,00	10976,00	1711,00

Taula 11. Desviacions del cost dels rols

Font: Elaboració pròpia

Les taules 10 i 11 ens donen una mica més de context dels llocs on s'han produït les desviacions més grans. Era fàcil preveure que la tasca "Afegir altra informació a BD" seria la que més es desviés, ja que en un principi estava catalogada com a risc. Aquest també és el motiu per el qual la desviació total dels rols és més gran que la de les tasques, no es van assignar les hores al programador de back-end per aquesta tasca, pero si es va considerar un cost per al risc, fet que comporta que el rol de programador de back-end tingui el desviament més elevat de tots els rols.

El que destaca també és el programador de front-end, que té un desviament del 49,33%. Això clarament denota la falta d'experiència a l'hora de fer planificacions d'aquest estil, ja que es van subestimar algunes de les històries d'usuari del front-end, que al final és la mateixa conclusió que ja s'havia extret.

Per altra banda, la desviació negativa més notable la té l'arquitecte, d'un 15%. Mirant la taula 10 s'observa que ve donada perquè els dissenys del front-end i el back-end van resultar ser més senzills de l'esperat.

Per concloure, penso que donades les causes explicades anteriorment, la desviació podria haver estat molt més elevada i altres tasques podrien haver vist alterada la seva qualitat. Al final, l'única cosa que no s'ha fet han estat els serveis web per al back-end, que des d'un principi es va dir que es farien al final del projecte si s'acabava amb un superàvit temporal. Totes les altres tasques i els seus requisits s'han fet i s'ha comprovat que passen una sèrie de tests i mètodes de validació per tal de garantir un mínim de qualitat a la feina feta.

Donat que aquesta és la primera vegada que he fet una planificació d'un projecte d'aquesta envergadura, el fet d'obtenir aquestes desviacions em servirà per no tornar a cometre el mateix error en projectes futurs, pel que crec que ha sigut una bona experiència d'aprenentatge.

12.2. Canvis de les històries d'usuari

La gran majoria de les històries d'usuari no s'han vist alterades durant el transcurs del desenvolupament del projecte, però algunes sí que han vist modificacions menors. Tot seguit es mostren les històries d'usuari canviades amb una breu explicació del canvi. Els canvis es mostren en cursiva.

Cas d'ús	#8 Veure equip	Actor principal	Usuari
Precondicions	-		
Disparador	L'usuari vol veure la informació d'un equip en concret		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol veure la informació d'un equip en concret.</div> <div>2. El sistema es comunica amb l'API, que retorna la informació de l'equip demanat.</div> <div>3. El sistema guarda la informació rebuda.</div> <div>4. El sistema redirigeix l'usuari a la vista "Equip", on es mostra la informació del equip demanat, <i>a més del llistat de les seves sèries properes i acabades.</i></div>			
Extensions			
<div>4a. <i>Alguna de les llistes retornades és buida.</i></div> <div>4a1. <i>El sistema indica a l'usuari que no s'ha trobat cap sèrie.</i></div>			

S'ha decidit que seria bona idea que es mostressin també les sèries de l'equip.

Cas d'ús	#8 Veure jugador	Actor principal	Usuari
Precondicions	-		
Disparador	L'usuari vol veure la informació d'un jugador en concret		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol veure la informació d'un jugador en concret.</div> <div>2. El sistema es comunica amb l'API, que retorna la informació del jugador demanat.</div> <div>3. El sistema guarda la informació rebuda.</div> <div>4. El sistema redirigeix l'usuari a la vista "jugador ", on es mostra la informació del jugador demanat, a més del llistat de les seves sèries properes i acabades.</div>			
Extensions			
<div>4a. Alguna de les llistes retornades és buida.</div> <div>4a1. El sistema indica a l'usuari que no s'ha trobat cap sèrie.</div>			

S'ha decidit que seria bona idea que es mostressin també les sèries del jugador.

Cas d'ús	#9 Iniciar sessió amb Google	Actor principal	Usuari
Precondicions	L'usuari no ha iniciat sessió en el sistema.		
Disparador	L'usuari vol iniciar sessió en el sistema.		
Escenari principal d'èxit			
<div>1. L'usuari indica al sistema que vol iniciar sessió amb Google.</div> <div>2. El sistema es comunica amb el servei de Google, que retorna el ID del usuari demanat.</div> <div>3. El sistema es comunica amb l'API, que retorna la informació del usuari demanat.</div> <div>4. El sistema guarda aquesta informació.</div> <div>5. El sistema crea una sessió per a aquest usuari.</div> <div>6. El sistema redirigeix l'usuari a la vista "Perfil".</div>			
Extensions			
<div>3a. No s'ha trobat cap usuari amb el ID especificat.</div> <div>3a1. El sistema es comunica amb el servei de Google, que retorna les dades bàsiques de l'usuari demanat.</div> <div>3a2. El sistema es comunica amb l'API, que crea aquest nou usuari a la base de dades</div> <div>3a3. Es torna al pas 3.</div>			

Al principi no hi havia precondició, ja que la idea era que es pogués iniciar sessió en un altre compte sense necessitat de primer tancar sessió, però es va decidir que tenia més sentit i era més intuïtiu d'aquesta manera.

13. Informe de sostenibilitat

En aquest capítol es du a terme l'informe de sostenibilitat del projecte. En la secció 13.1 hi ha una autoavaluació general sobre la sostenibilitat. Després, estan les autoavaluacions de les diferents dimensions de la sostenibilitat: Dimensió econòmica (secció 13.2), dimensió social (secció 13.3) i dimensió ambiental (secció 13.4). Per últim, en la secció 13.5, es mostra una taula resum de l'autoavaluació de la sostenibilitat d'aquest projecte.

13.1. Autoavaluació

Fent l'enquesta que ens van proporcionar els professors de GEP, m'he adonat que potser no sé tant com em pensava sobre certs camps de la sostenibilitat, mentre que n'hi ha d'altres que domino força bé.

Primer de tot, cal dir que algunes de les preguntes no acaben d'aplicar-se del tot en aquest projecte. Un exemple seria si es tracta de millorar el bé comú de la societat. Òbviament, no l'empitjora, i l'única millora real és que les persones interessades podran accedir a informació sobre els e-sports de manera més fàcil i còmode, però crec que cal recordar que en l'àmbit global de la societat, els *e-sports*, igual que els esports tradicionals, no són importants. Important és un descobriment que ajuda a curar una malaltia, no unes persones jugant a videojocs o donant puntades de peu a una pilota.

Dit això, penso que sí que sóc capaç de valorar correctament l'impacte que un projecte pot tenir a l'hora de millorar la societat. També comprenc bastant bé les causes, conseqüències i solucions que es donen normalment respecte de la problemàtica social, econòmica i ambiental, i sóc capaç d'analitzar-les i relacionar el problema amb altres problemes i solucions existents.

Comprend la importància de les TIC en la sostenibilitat del planeta, ja que és un tema molt important que es porta repetint bastant en els darrers anys, i conec la majoria dels conceptes relacionats amb la sostenibilitat, tot i que hauria d'aprofundir en la problemàtica associada a alguns com pot ser la justícia social, i altres són nous per a mi, com és el cas dels principis deontològics.

Clarament hauria de millorar certes coses, com he dit abans la majoria de coses em sonen en l'àmbit teòric, però no sabia aplicar-les en l'àmbit pràctic, com pot ser aportar les idees adequades per tal de millorar la sostenibilitat del projecte, saber aplicar les mesures adients, o ser capaç d'incloure indicadors per estimar la contribució del projecte a la millora de la sostenibilitat.

En resum, opino que conec prou bé els diferents conceptes relacionats amb la sostenibilitat, però crec que em falta pràctica a l'hora d'aplicar aquests conceptes per tal d'aconseguir que els meus projectes siguin més sostenibles.

13.2. Dimensió econòmica

En la gestió econòmica hem vist que existeix una àmplia valoració dels costos propis del projecte, tant dels derivats dels recursos humans com dels costos genèrics. A més, es fa una valoració dels possibles imprevistos i es té en compte un percentatge de contingència.

Aquests costos s'han intentat reduir el màxim possible, i en alguns casos s'han triat recursos de software gratuïts abans de productes similars de pagament, pel que resultaria difícil aconseguir fer el mateix projecte per un preu menor.

S'ha intentat reduir el màxim possible el temps dedicat a les diferents tasques, agafant llibreries ja existents i fent diverses tasques de manera concurrent.

Actualment, els costos d'altres projectes similars són semblants als d'aquest projecte. En els casos d'aplicacions d'empreses privades, no es poden saber amb exactitud aquests costos, però molt probablement s'assemblen als d'aquest projecte, utilitzant uns rols i sistemes similars. En altres casos, s'han fet de manera voluntària i sense ànim de lucre, pel que no es pot competir amb aquestes solucions en qüestions de cost.

Pel que fa a la viabilitat econòmica del projecte, s'ha contemplat llançar-ho al mercat una vegada es finalitzi el desenvolupament de l'aplicació. En el moment de redactar aquest informe, no hi ha cap aplicació per a dispositius mòbils similar que estigui a l'altura de les expectatives dels usuaris, pel que es considera que aquest projecte és viable i serà competitiu en el mercat.

13.3. Dimensió social

Ja fa bastants anys que la situació del país és bastant dolenta, havent una taxa d'atur molt elevada. Tot i això, el sector de la informàtica és una excepció a aquesta norma; la gran majoria de gent amb un mínim de formació en aquest sector és capaç de trobar feina sense gaires problemes a causa de la gran demanda de professionals que existeix, més encara en el desenvolupament d'aplicacions per a dispositius mòbil.

Aquest projecte no perjudica a ningú, simplement facilita a aquells usuaris interessats en els *e-sports* a trobar dades de manera més fàcil i ràpida.

En l'àmbit personal, opino que aquest projecte m'ajudarà molt a millorar com a professional i a dominar els diferents aspectes necessaris per dur a terme un projecte.

En l'apartat d'estat de l'art, s'ha conclòs que, tot i que hi ha pàgines webs bones d'aquest tema, no hi ha cap aplicació mòbil que estigui a l'altura de les expectatives que pugui tenir un usuari, pel que realment hi ha una necessitat real de realitzar aquest projecte.

13.4. Dimensió ambiental

Actualment i per desgràcia, vivim en una societat on moltes vegades es menysprea l'impacte ambiental a favor d'aconseguir més diners. Aquest projecte és un projecte de desenvolupament de software, pel que realment no es pot fer molta cosa per millorar aquest impacte ambiental en l'àmbit global, que també és cert per solucions actuals del problema, però això no vol dir que no es pugui intentar reduir el màxim possible dins de les nostres possibilitats.

Com s'ha explicat abans, es reutilitzaran llibreries de tercers per tal de reduir la quantitat de feina i, per tant, l'impacte ambiental del projecte, ja que s'utilitzaran els recursos de hardware que consumeixen energia durant menys estona. També es faran tots els dissenys de l'aplicació (mock-ups) utilitzant paper reciclat.

Com s'ha esmentat abans, aquest projecte pretén facilitar la consulta de dades sobre els *e-sports* respecte de les solucions existents, és a dir, disminuir el temps de cerca, pel que es considera que un usuari que utilitzi aquesta aplicació tindrà un impacte menor al d'un usuari que utilitzi altres mètodes per trobar aquesta informació.

13.5. Matriu de sostenibilitat

En la taula 12 es mostra la matriu de sostenibilitat del projecte, en la que s'especifica la valoració autoavaluada dels diferents àmbits de la sostenibilitat del projecte.

Sostenibilitat	Econòmica	Social	Ambiental	Total
Planificació	Viabilitat econòmica	Millora en la qualitat de vida	Anàlisi de recursos	
Valoració	9	8	8	25/30

Taula 12. Matriu de sostenibilitat
Font: Elaboració pròpia

14. Conclusions

En aquest capítol es troben les conclusions del projecte. Primer, es parla sobre quines competències tècniques s'han treballat durant el projecte (secció 14.1), després s'expliquen diverses funcionalitats que es podrien desenvolupar per al sistema en un futur (secció 14.2) i per últim es redacten les meves conclusions personals respecte al sistema i al projecte com a autor del projecte (secció 14.3).

14.1. Competències tècniques treballades

Tot seguit es mostren les diferents competències tècniques que s'han treballat en el transcurs d'aquest projecte, i la seva relació amb l'especialitat d'enginyeria del software.

CES1.1: Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics. [Bastant]

Aquesta és probablement la competència tècnica més treballada en aquest projecte. En aquesta memòria s'ha dit que "L'objectiu principal d'aquest projecte és desenvolupar una aplicació d'Android en què es mostri informació i estadístiques sobre diferents e-sports, a més d'altres funcionalitats com poder comentar les diferents partides", pel que clarament s'ha desenvolupat un sistema software complex, que comporta una aplicació Android i tot un back-end on emmagatzemar les dades. A més, aquest sistema s'ha mantingut durant el transcurs del projecte i també s'han avaluat diferents sistemes de software en l'estat de l'art. Aquesta competència és una de les més bàsiques de l'enginyeria del software, ja que es tracta de desenvolupar el mateix sistema.

CES1.2: Donar solució a problemes d'integració en funció de les estratègies, dels estàndards i de les tecnologies disponibles. [Bastant]

En l'enginyeria del software és important trobar solucions a diferents problemes amb els recursos disponibles. En aquest cas, s'ha integrat una API externa al back-end del sistema i, com que no donava tota la informació necessària, s'han extret les dades d'altres llocs en funció de les tecnologies disponibles, més concretament, l'HTML Scraping. A més, s'ha dissenyat, desenvolupat i integrat una API per comunicar el front-end amb el back-end, ja que era l'estratègia més viable.

CES1.3: Identificar, avaluar i gestionar els riscos potencials associats a la construcció de software que es poguessin presentar. [Una mica]

Un bon enginyer de software ha de tenir present els diferents riscos que poguessin impedir o alentir la construcció del software en qüestió. En aquest projecte i en aquesta memòria, s'han definit i descrit clarament els possibles obstacles i s'han proposat solucions que s'han seguit en el cas que el risc es materialitzés. Aquests riscos s'han tingut en compte durant tot el transcurs del projecte: durant la planificació inicial, la planificació temporal, la gestió econòmica i durant el desenvolupament del sistema, sempre tenint presents les solucions proposades per no perdre temps i recursos en el cas que es presentés algun dels riscos.

CES1.4: Desenvolupar, mantenir i avaluar serveis i aplicacions distribuïdes amb suport de xarxa. [Una mica]

Algunes parts del sistema desenvolupat i mantingut depenen de tenir suport de xarxa. L'exemple més clar és el back-end, que es comunica amb l'API externa de PandaScore i en serveis externs mitjançant una connexió a Internet per extreure les dades faltants. A més, el front-end es comunica amb l'API amb crides asíncrones, pel que és necessari seguir un protocol de com afrontar aquestes crides.

CES1.5: Especificar, dissenyar, implementar i avaluar bases de dades. [Bastant]

Una de les bases de l'enginyeria del software és el disseny de bases de dades. En aquest projecte el back-end ha estat una part molt important, de fet, més important que el front-end, almenys durant el desenvolupament. S'han avaluat diferents models de base de dades com poden ser relacionals, basades en documents o en grafs i s'ha decidit per una basada en documents. Aquests documents s'han especificat, dissenyat i implementat, tal com es mostra en aquesta memòria.

CES1.6: Administrar bases de dades (CIS4.3) [Bastant]

Clarament, el següent pas de dissenyar i implementar una base de dades és administrar-la. La base de dades s'actualitza regularment afegint i modificant els documents pertinents. També s'actualitza i afegeix informació relativa als usuaris des del front-end del sistema, com pot ser editar el nom d'usuari o afegir un equip a preferits.

CES1.7: Controlar la qualitat i dissenyar proves en la producció de software. [Bastant]

Un dels aspectes fonamentals de l'enginyeria del software és tenir un control sobre la qualitat del sistema, pel que s'han definit i executat diferents mètodes de validació que inclouen tests automàtics, tests manuals de les funcionalitats i tests d'usabilitat i dels requisits no funcionals amb usuaris externs al desenvolupament del projecte. Aquests tests garanteixen la qualitat del sistema i que funcioni correctament.

CES1.9: Demostrar comprensió en la gestió i govern dels sistemes software. [Una mica]

En aquest projecte s'ha demostrat una comprensió en la gestió del sistema software que s'ha desenvolupat. La prova és que s'han seguit correctament la gran majoria de requisits, i el sistema funciona correctament en tots els casos.

CES2.1: Definir i gestionar els requisits d'un sistema software. [Bastant]

És important tenir ben definits els requisits del sistema a desenvolupar. En aquest projecte vaig definir primer les tasques principals del projecte, que després vaig dividir en requisits. Pels requisits funcionals vaig definir les precondicions, el disparador, l'escenari principal d'èxit i les diferents extensions del cas d'ús, mentre que pels no funcionals vaig definir la descripció del requisit i el criteri d'acceptació d'aquest. Durant el desenvolupament del projecte, vaig anotar els requisits funcionals que calia modificar i vaig fer tests d'usabilitat per a garantir que els no funcionals es complien.

14.2. Treball futur

Aquest és un sistema amb moltes possibilitats, pràcticament infinites, pel que hi ha moltíssimes coses que es poden fer d'acord amb aquest sistema. A continuació, explicaré algunes de les que m'agradaria fer fora de l'abast d'aquest projecte.

- **Desenvolupament de serveis web per al back-end.** Aquesta clarament havia d'estar, ja que forma part de l'abast inicial del projecte. Com s'explica en l'apartat 5.1.10, es tracta de desenvolupar alguns serveis web (és a dir, una pàgina web) per a facilitar algunes tasques a l'administrador de la base de dades, com pot ser que rebí notifikacions de les inconsistències detectades, o que pugui modificar la base de dades des d'una interfície d'usuari usable, en comptes de directament des de la web on està allotjada o mitjançant scripts de Python.
- **Sistema de notifikacions.** Al principi, aquesta era una de les funcionalitats que tindria l'aplicació, però finalment es va descartar perquè no es sabia encara la viabilitat d'iniciar sessió al sistema i es va considerar que era una funcionalitat "extra" que no era necessària incloure a l'abast del projecte. Bàsicament, els usuaris rebrien notifikacions mitja hora abans de començar una partida dels seus equips preferits, i una altra amb el resultat si no tenien el mode sense spoilers activat.
- **Veure partides en diferit mitjançant l'API de Youtube.** Aquesta també es va pensar inicialment, i, fins i tot, es poden veure els dissenys en l'annex 1, però es va descartar perquè era bastant inviable. El fet d'integrar l'API de Youtube no suposava cap problema, el problema venia a l'hora de trobar els vídeos corresponents. Durant les primeres proves, es va estimar que només es podia trobar de manera automàtica el 35% de les partides (només algunes de LoL), pel que es va deixar com a treball futur si es trobava alguna manera viable de trobar-les.
- **Millorar l'automatització de les actualitzacions de la base de dades.** Com s'ha anat dient al llarg de la documentació, hi ha hagut molts problemes relacionats amb el back-end, però el més important són les dificultats per a automatitzar les actualitzacions de la base de dades. S'intentarà buscar alguna manera efectiva i viable de fer-ho.
- **Afegir funcionalitats socials.** S'ha dit en aquesta memòria que es vol transmetre la sensació de comunitat quan s'utilitza aquesta aplicació. Ara mateix es pot iniciar sessió i comentar les partides, que és la base per a mantenir una comunitat, però es poden fer més coses. Això és una cosa que s'ha de pensar prudentment, ja que he vist de primera mà com algunes pàgines web han mort per afegir funcionalitats socials que no han estat ben acceptades pels usuaris, però algunes funcionalitats podrien incloure: votar al millor jugador de la partida, afegir noves maneres d'iniciar sessió o permetre als usuaris votar a qui creuen que guanyarà una partida abans que comenci.

14.3. Conclusions personals

Ara fa un any, no tenia gens clar com afrontar el TFG. Per més que hi pensés, no hi havia manera de trobar un tema suficientment bo que em motivés a fer un treball d'aquesta envergadura, pel que vaig suposar que el millor seria fer-ho en una empresa.

Però una nit, parlant amb un amic, em va recordar que el Barça estava jugant, pel que vaig mirar el resultat en una aplicació mòbil. Mentre s'obria l'aplicació, pensava que estaria bé que existís alguna aplicació similar d'*e-sports* ja que no hi havia cap decent, i va ser el moment en el qual vaig trobar el tema perfecte per fer jo sol el projecte: desenvolupar una aplicació per veure els resultats de partides d'*e-sports*.

En les setmanes següents, amb aquesta premissa al cap, vaig començar a llistar totes les coses que a mi m'agradaria que tingués una aplicació d'aquest estil, fins que vaig arribar a l'objectiu principal del projecte que es mostra a l'inici d'aquesta memòria: "L'objectiu principal d'aquest projecte és dissenyar i desenvolupar una aplicació Android que faciliti la cerca de dades i estadístiques sobre competicions, equips i jugadors de diferents e-sports i disposi de diverses funcionalitats socials, com comentar les partides."

Durant aquest període de temps, l'abast del projecte era immens, ja que volia fer moltes coses, però a mesura que passava el temps i afegia coses que considerava bàsiques, em vaig adonar que potser era millor tenir un abast més reduït però que estigués ben fet, i deixar aquestes funcionalitats "extra" per al treball futur o per si sobrava temps. Aquesta és una de les principals conclusions que extrec d'aquest projecte: Moltes vegades no es pot fer tot el que es vol, però això no vol dir que el resultat sigui pitjor i, fins i tot, de vegades és millor.

Pel que fa a l'objectiu principal del projecte, s'ha complert. S'ha desenvolupat una aplicació Android funcional que permet visualitzar dades, resultats i estadístiques sobre competicions de dos e-sports diferents. Així mateix també s'han assolit els altres objectius marcats a la secció 3.1 de la documentació: Triar i integrar una API externa, fer el back-end i fer l'aplicació client d'Android, pel que considero que el projecte ha estat un èxit.

Abans de començar aquest projecte jo ja tenia una mica d'experiència desenvolupant en Android, gràcies a l'assignatura de PES (Projectes d'Enginyeria del Software) de la FIB, on havíem de desenvolupar una aplicació Android en un grup de 5 persones. En aquella assignatura jo em dedicava a ser programador de front-end, pel que moltes de les coses necessàries ja les sabia, però també crec que he après molt i he millorat els coneixements previs que ja tenia. Per exemple, crec que he utilitzat millor el patró MVC i he aprofundit en alguns conceptes útils del desenvolupament en Android, tant en l'àmbit teòric com en el pràctic, com pot ser utilitzar els fragments i els adaptadors correctament.

On realment he millorat i he après molt ha sigut amb el back-end. Abans de començar jo tenia uns coneixements "normals" de Python i molt bàsics de MongoDB i bases de dades en general, ja que, sincerament, no em cridava molt l'atenció. Després d'acabar el projecte, puc

dir que m'he adonat que treballar amb bases de dades és molt més interessant i gratificant del que em pensava i, fins i tot, puc arribar a estar interessat en ofertes de feina per a programador de back-end, cosa que considerava impensable no fa tant.

Aquest ha estat el projecte més gran que he fet mai, només el projecte de PES que he esmentat abans s'apropa, i així i tot, aquest ha sigut més gran. A més aquest projecte l'he fet jo sol, amb consells de la meva directora, els meus pares i els meus amics. Clarament no tot ha sortit perfecte tal com jo m'esperava i volia ni totes les idees que tenia eren bones; com ja he dit abans, moltes coses vaig decidir ni incloure-les a l'abast, i fins i tot algunes coses que sí que vaig incloure no s'han fet, com s'ha vist a l'apartat 12 de seguiment del projecte. Aquesta és una altra cosa que he après en aquest projecte, vaig ser massa optimista en les estimacions de temps, donats els meus coneixements limitats de desenvolupament de back-end.

Tot i això, realment penso que el projecte s'ha finalitzat correctament i estic molt content de com ha quedat. De tota la vida m'han agradat molt les competicions esportives de tota mena i analitzar les estadístiques de diferents jugadors i equips, pel que sempre he treballat de forma molt optimista en aquest projecte i en cap moment m'he sentit desanimat quan m'he encallat ni mai he perdut la motivació per a treballar en el projecte ni he sentit que ho estava fent totalment per obligació, fet que ha estat clau per a garantir el compliment dels objectius del projecte amb un mínim de qualitat.

No crec que aquest projecte acabi amb aquesta entrega, encara queden coses per fer que he explicat més detalladament en el treball futur. He estat pensant aquests últims dies que fer amb el projecte després d'acabar el TFG. Intentar publicar-ho en l'estat actual és una opció però no la millor, què és pagar els 300 € al mes de l'API de PandaScore per a garantir actualitzacions instantànies i no haver de dependre de pàgines externes. Potser contactar amb alguna empresa que l'interessi? Encara no estic del tot segur quina és la millor opció, però clarament queden coses per fer.

15. Referències

- [1] - [En línia] - What Exactly Is Agile? A Definition of Agile Project Management - Andrew Conrad - Data de consulta: 20/9/19
<https://blog.capterra.com/definition-of-agile-project-management/>
- [2] - [En línia] - Android (operating system) - Data de consulta: 20/9/19
[https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [3] - [En línia] - Wikipedia:About - Data de consulta: 20/9/19
<https://en.wikipedia.org/wiki/Wikipedia:About>
- [4] - [En línia] - Primer Torneo de eSport - Hugo Lecourt - Data de consulta: 20/9/19
<https://bequipe.com/blogs/infos/first-esport-tournament>
- [5] - [En línia] - 427 million people will be watching esports by 2019, reports Newzoo - Darren Rovell - Data de consulta: 20/9/19
https://www.espn.com/esports/story/_/id/15508214/427-million-people-watching-esports-2019-reports-newzoo
- [6] - [En línia] - Worlds 2018 — 200 million viewers at once - Data de consulta: 20/9/19
<https://escharts.com/blog/worlds-2018-final>
- [7] - [En línia] - LEC SUMMER FINALS VIEWERSHIP UPDATE - Alberto "Tiroless" Guerrero - Data de consulta: 20/9/19
<https://eu.lolesports.com/en/articles/lec-summer-finals-viewership-update>
- [8] - [En línia] - Splyce Vipers gana la final más vista de la historia de la Superliga Orange de League of Legends - Data de consulta: 20/9/19
<https://www.esportsbureau.com/slo-lol-splyce-vipers/>
- [9] - [En línia] - The International 2019 - Data de consulta: 21/9/19
https://liquipedia.net/dota2/The_International/2019
- [10] - [En línia] - LoL Esports - Data de consulta: 21/9/19
<https://watch.lolesports.com/>
- [11] - [En línia] - liquipedia - Data de consulta: 21/9/19
<https://liquipedia.net/>
- [12] - [En línia] - Leaguepedia - Data de consulta: 21/9/19
https://lol.gamepedia.com/League_of_Legends_Esports_Wiki
- [13] - [En línia] - Reddit - Data de consulta: 21/9/19
<https://www.reddit.com/>
- [14] - [En línia] - Eventvods - Data de consulta: 21/9/19
<https://eventvods.com/>

- [15] - [En línea] - HLTV - Data de consulta: 21/9/19
<https://www.hltv.org/>
- [16] - [En línea] - Tendencias del mercado laboral - Data de consulta: 5/10/19
https://www.michaelpage.es/sites/michaelpage.es/files/PG_ER_IT.pdf
- [17] - [En línea] - Conceptual Data Model - Data de consulta: 12/12/19
<https://www.sciencedirect.com/topics/computer-science/conceptual-data-model>
- [18] - Robertson, S.; Robertson, J. *Mastering the requirements process : getting requirements right*. Upper Saddle River : Addison-Wesley, cop. 2013. 3a edició. ISBN 9780321815743
- [19] - [En línea] - MongoDB - Data de consulta: 13/12/19
<https://www.mongodb.com/>
- [20] - [En línea] - MongoDB Atlas - Data de consulta: 13/12/19
<https://www.mongodb.com/cloud/atlas>
- [21] - [En línea] - PandaScore - Data de consulta: 13/12/19
<https://pandascore.co/>
- [22] - [En línea] - Heroku - Data de consulta: 13/12/19
<https://www.heroku.com/>
- [23] - [En línea] - Non-relational data and NoSQL - Data de consulta: 13/12/19
<https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data>
- [24] - [En línea] - Non-relational data and NoSQL - Data de consulta: 14/12/19
<https://developer.android.com/reference/android/view/View>
- [24] - [En línea] - View - Data de consulta: 14/12/19
<https://developer.android.com/reference/android/view/View>
- [25] - [En línea] - Activity - Data de consulta: 14/12/19
<https://developer.android.com/reference/android/app/Activity.html>
- [26] - [En línea] - Fragment - Data de consulta: 14/12/19
<https://developer.android.com/reference/android/app/Fragment.html>
- [27] - [En línea] - The DCI Architecture: A New Vision of Object-Oriented Programming - Data de consulta: 17/12/19
https://www.artima.com/articles/dci_vision.html
- [28] - [En línea] - Patrón Modelo Vista Controlador - Data de consulta: 17/12/19
<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122>
- [29] - [En línea] - Singleton - Data de consulta: 17/12/19
<https://refactoring.guru/design-patterns/singleton>
- [30] - [En línea] - Python - Data de consulta: 28/12/19

<https://www.python.org/>

[31] - [En línea] - About Python - Data de consulta: 28/12/19

<https://www.python.org/about/>

[32] - [En línea] - Python Programming - Data de consulta: 28/12/19

https://en.wikibooks.org/wiki/Python_Programming/Version_history

[33] - [En línea] - pandascore - Data de consulta: 28/12/19

<https://pypi.org/project/pandascore/>

[34] - [En línea] - PyMongo - Data de consulta: 28/12/19

<https://api.mongodb.com/python/current/>

[35] - [En línea] - SeleniumHQ - Data de consulta: 28/12/19

<https://selenium.dev/>

[36] - [En línea] - urllib - Data de consulta: 28/12/19

<https://docs.python.org/3/library/urllib.html>

[37] - [En línea] - Pillow · PyPI - Data de consulta: 28/12/19

<https://pypi.org/project/Pillow/>

[38] - [En línea] - lxml - Data de consulta: 28/12/19

<https://lxml.de/3.7/>

[39] - [En línea] - Flask - Data de consulta: 28/12/19

<https://www.palletsprojects.com/p/flask/>

[40] - [En línea] - Java - Data de consulta: 28/12/19

<https://www.java.com/es/>

[41] - [En línea] - Retrofit - Data de consulta: 28/12/19

<https://square.github.io/retrofit/>

[42] - [En línea] - Glide - Data de consulta: 28/12/19

<https://github.com/bumptech/glide>

[43] - [En línea] - Google Play Services - Data de consulta: 28/12/19

<https://developers.google.com/android/guides/setup>

[44] - [En línea] - Extensible Markup Language (XML) - Data de consulta: 28/12/19

<https://www.w3.org/XML/>

[45] - [En línea] - Introduction to XML - Data de consulta: 28/12/19

https://www.w3schools.com/xml/xml_what_is.asp

[46] - [En línea] - Descripción general de AndroidX - Data de consulta: 28/12/19

<https://developer.android.com/jetpack/androidx>

[47] - [En línea] - Android Jetpack - Data de consulta: 28/12/19

<https://developer.android.com/jetpack>

[48] - [En línea] - Cómo migrar a AndroidX - Data de consulta: 28/12/19

<https://developer.android.com/jetpack/androidx/migrate>

[49] - [En línea] - PyCharm - Data de consulta: 28/12/19

<https://www.jetbrains.com/pycharm/>

[50] - [En línea] - Android Studio - Data de consulta: 28/12/19

<https://developer.android.com/studio>

[51] - [En línea] - Ionic - Data de consulta: 28/12/19

<https://ionicframework.com/>

[52] - [En línea] - Heroku - Data de consulta: 28/12/19

<https://heroku.com/>

[53] - [En línea] - HTML Scapping - Data de consulta: 28/12/19

<https://docs.python-guide.org/scenarios/scrape/>

Llicència: <https://creativecommons.org/licenses/by-nc-sa/3.0/>

[54] - [En línea] - PandaScore | Pricing - Data de consulta: 30/12/19

<https://pandascore.co/pricing>

[55] - [En línea] - 2019 LoL KeSPA Cup - Data de consulta: 30/12/19

https://lol.gamepedia.com/2019_LoL_KeSPA_Cup/Match_History

[56] - [En línea] - CS:GO Results - Data de consulta: 30/12/19

<https://www.hltv.org/results?event=4450>

[57] - [En línea] - Ionic - Data de consulta: 28/12/19

<https://ionicframework.com/>

[58] - [En línea] - Ionic - Data de consulta: 28/12/19

<https://ionicframework.com/>

[59] - [En línea] - Ionic - Data de consulta: 28/12/19

<https://ionicframework.com/>

[60] - [En línea] - JUnit - Data de consulta: 05/01/20

<https://junit.org/junit4/>

[61] - [En línea] - CompletableFuture - Data de consulta: 05/01/20

<https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletableFuture.html>

[62] - [En línea] - Legal | Riot Games - Data de consulta: 09/01/20

<https://www.riotgames.com/en/legal>

[63] - [En línea] - API Terms & Conditions | Riot Games - Data de consulta: 09/01/20

<https://developer.riotgames.com/terms>

- [64] - [En línea] - Políticas | Riot Games - Data de consulta: 09/01/20
<https://developer.riotgames.com/policies/general>
- [65] - [En línea] - Información legal | Steam - Data de consulta: 09/01/20
<https://store.steampowered.com/legal/>
- [66] - [En línea] - Legal Information Institute - Data de consulta: 09/01/20
<https://www.law.cornell.edu/uscode/text/17/107>
- [67] - [En línea] - Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo - Data de consulta: 09/01/20
<https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX%3A32016R0679>
- [68] - [En línea] - mongoDB Features - Data de consulta: 09/01/20
<https://www.mongodb.com/cloud/atlas/faq>
- [69] - [En línea] - What is SSL, TLS and HTTPS | digicert - Data de consulta: 09/01/20
<https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>
- [70] - [En línea] - The MIT License - Data de consulta: 09/01/20
<https://opensource.org/licenses/MIT>
- [71] - [En línea] - APACHE LICENSE, VERSION 2.0 - Data de consulta: 09/01/20
<https://www.apache.org/licenses/LICENSE-2.0>
- [72] - [En línea] - The 3-Clause BSD License - Data de consulta: 09/01/20
<https://opensource.org/licenses/BSD-3-Clause>
- [73] - [En línea] - GPL - Data de consulta: 09/01/20
<https://www.gnu.org/licenses/gpl-3.0.html>
- [74] - [En línea] - PandaScore Terms and legal mentions - Data de consulta: 09/01/20
<https://pandascore.co/terms>
- [75] - [En línea] - Python License - Data de consulta: 09/01/20
<https://docs.python.org/3/license.html>
- [76] - [En línea] - Server Side Public License (SSPL) - Data de consulta: 09/01/20
<https://www.mongodb.com/licensing/server-side-public-license>
- [77] - [En línea] - Server Side Public License FAQ - Data de consulta: 09/01/20
<https://www.mongodb.com/licensing/server-side-public-license/faq>
- [78] - [En línea] - GNU Affero General Public License - Data de consulta: 09/01/20
<https://www.gnu.org/licenses/agpl-3.0.html>

Annexe 1. Dissenys inicials en paper

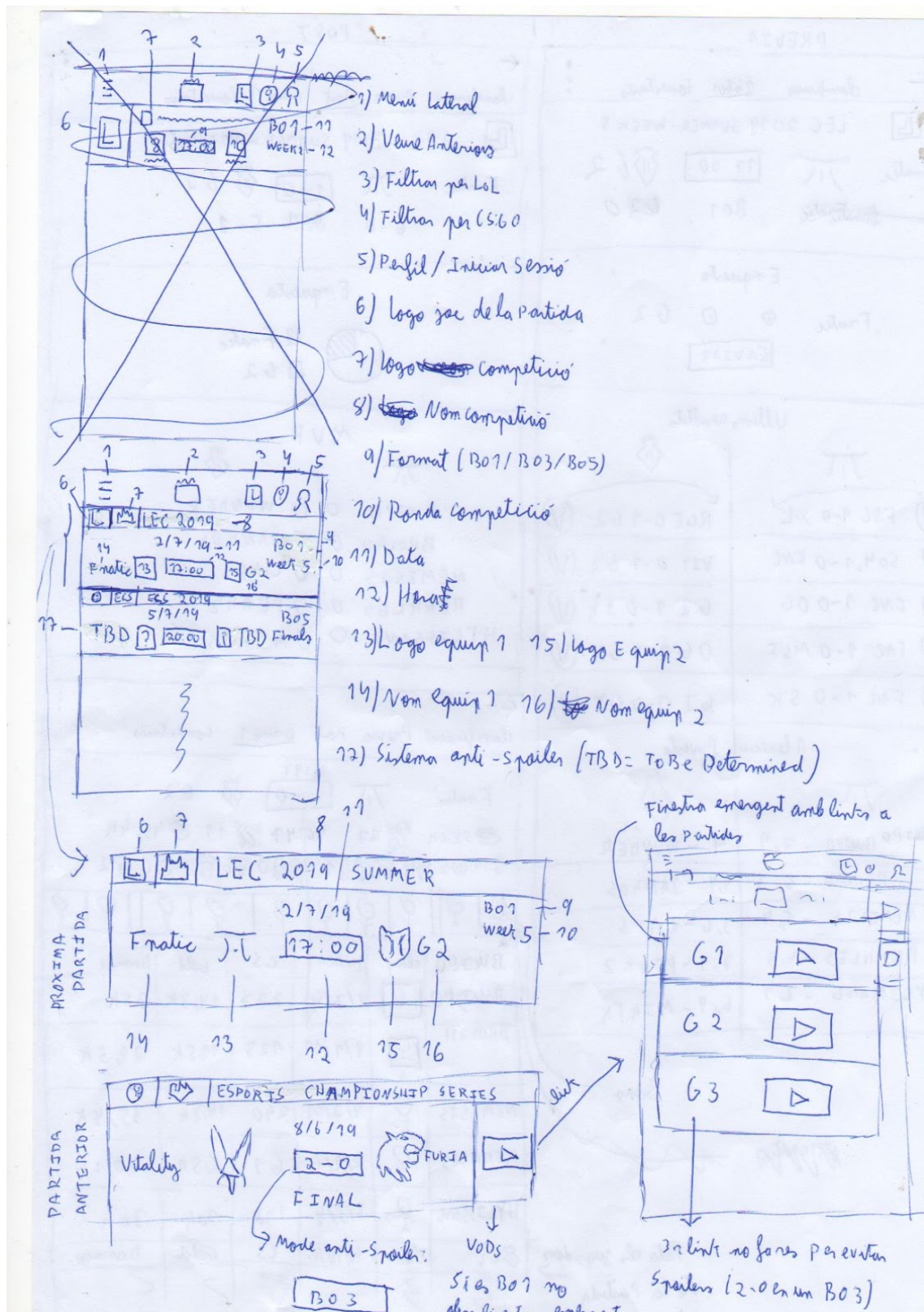


Figura 70. Disseny inicial de la pàgina principal

Font: Elaboració pròpia

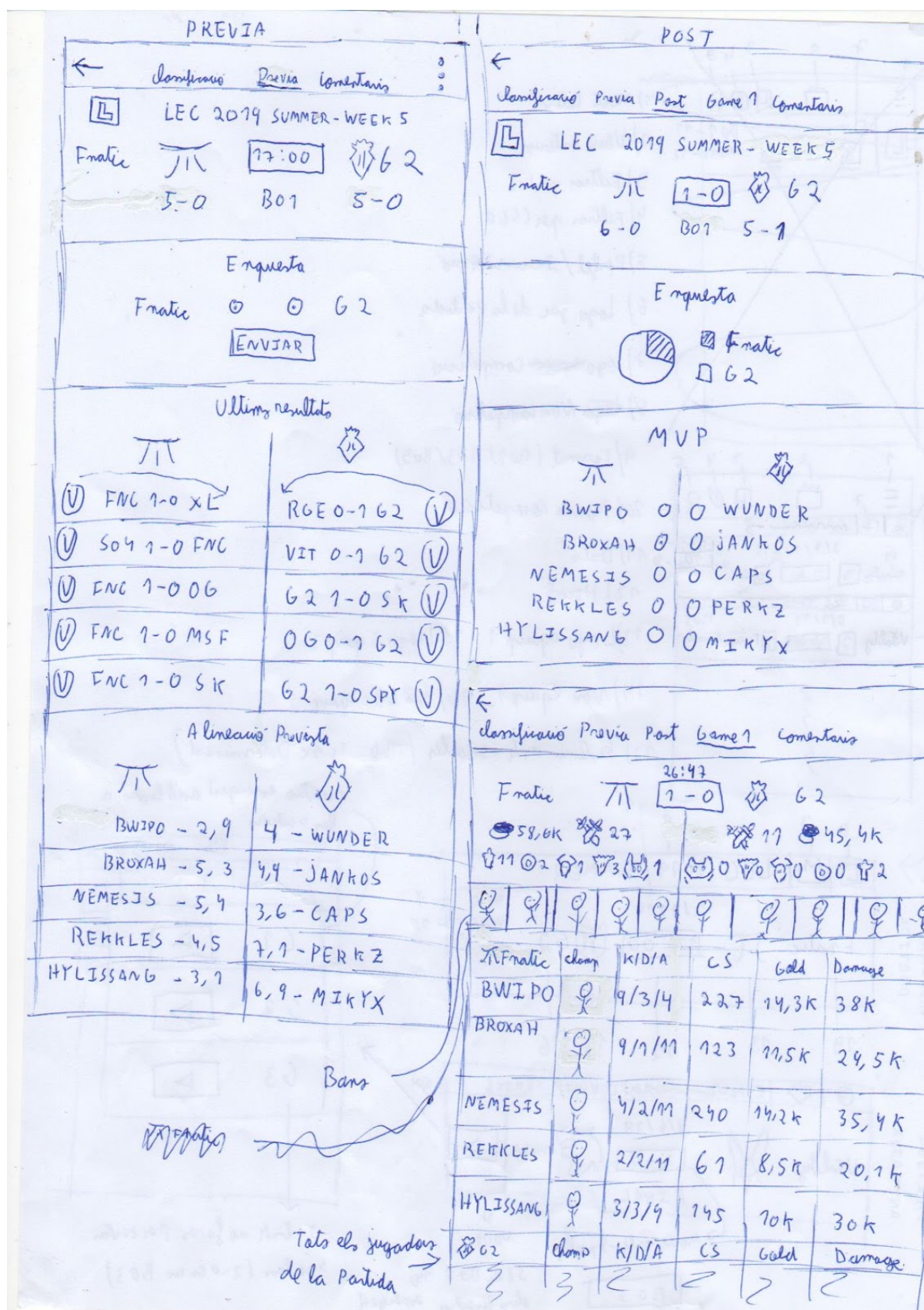


Figura 71. Disseny inicial de la vista d'una sèrie de LoL

Font: Elaboració pròpia

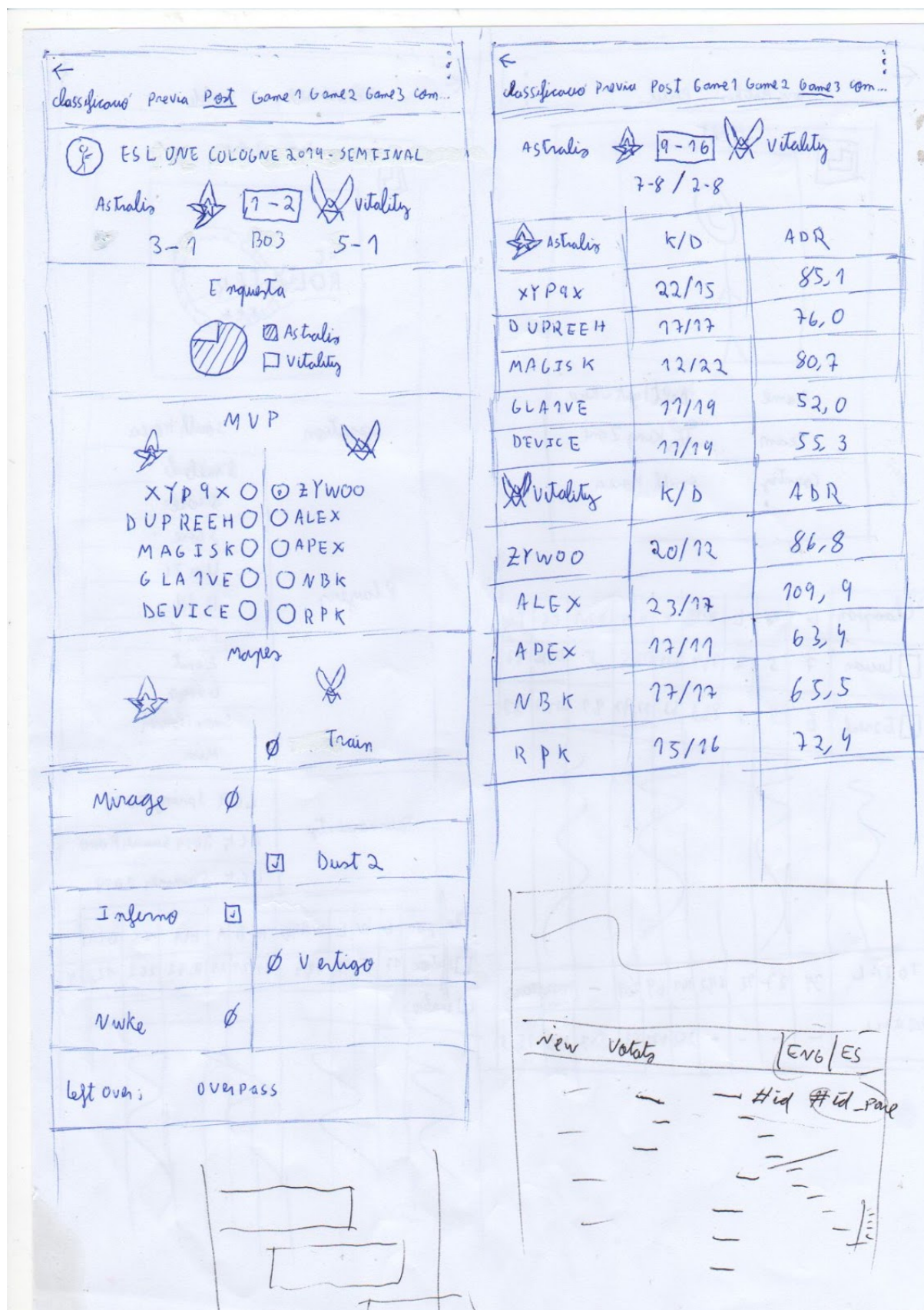


Figura 72. Disseny inicial de la vista d'una sèrie de CSGO i dels comentaris (abaix-dreta)

Font: Elaboració pròpia

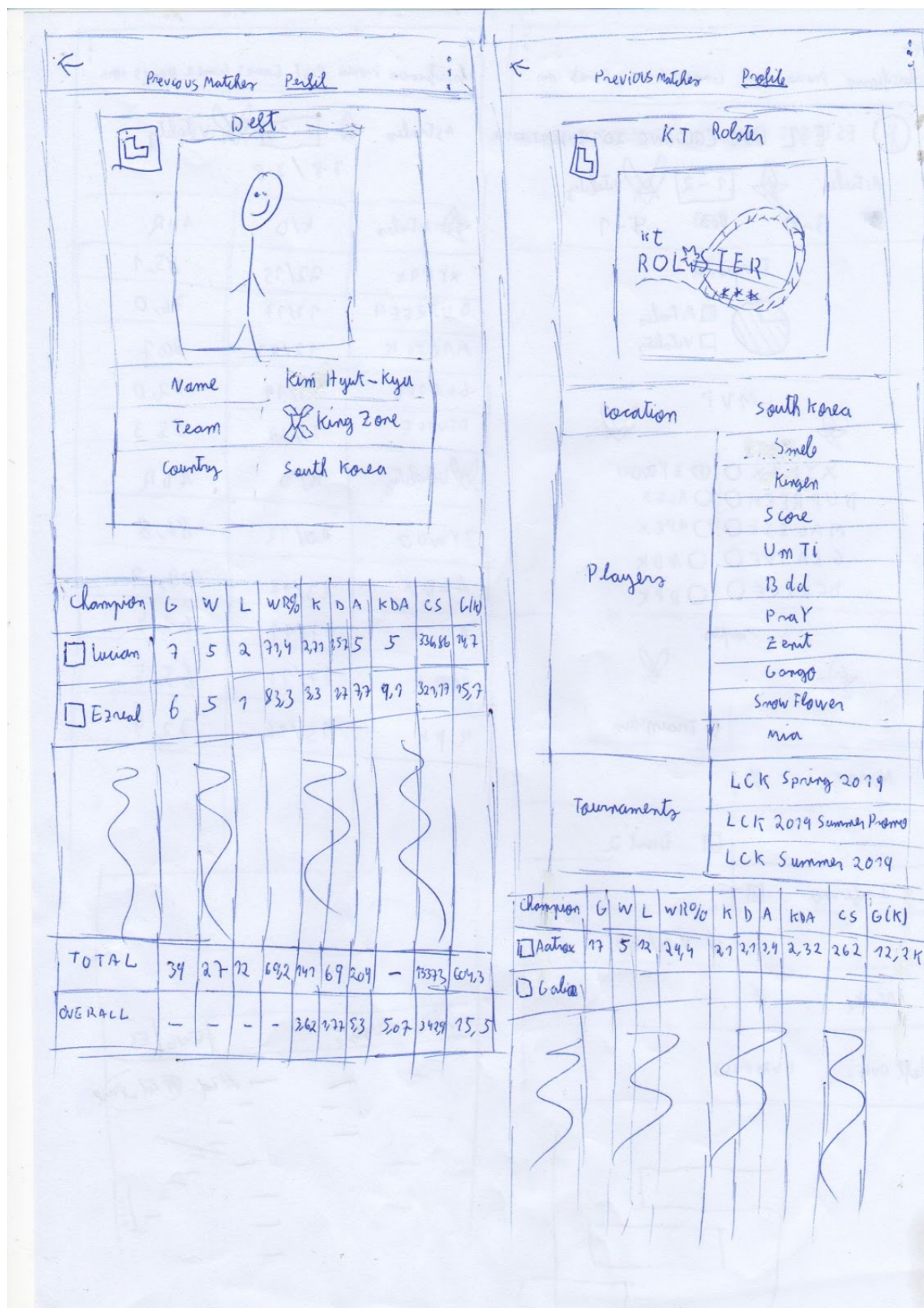


Figura 73. Disseny inicial de la vista d'un jugador (esquerra) i un equip (dreta)

Font: Elaboració pròpia